

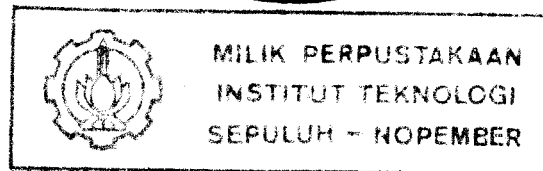
3100096007340

**INTERFACE AKUISISI 6 PARAMETER DATA METEOROLOGI  
DENGAN MINIMUM SISTEM 8088 YANG DIHUBUNGKAN  
PADA KOMPUTER IBM PC SECARA SERI**

PERPUSTAKAAN ITS	
Tgl. Terima	22 SEP 1994
Terima Dari	H
No. Agenda Prp.	2734



RSE  
621.398.1  
Sam  
E-1  
1994



MILIK PERPUSTAKAAN  
INSTITUT TEKNOLOGI  
SEPULUH - NOPEMBER

OLEH :

**DADANG SAMUEL**

**NRP. 291 220 1796**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1994**

**INTERFACE AKUISISI 6 PARAMETER DATA METEOROLOGI  
DENGAN MINIMUM SISTEM 8088 YANG DIHUBUNGKAN  
PADA KOMPUTER IBM PC SECARA SERI**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

**P a d a**

**Bidang Studi Elektronika**

**Jurusan Teknik Elektro**

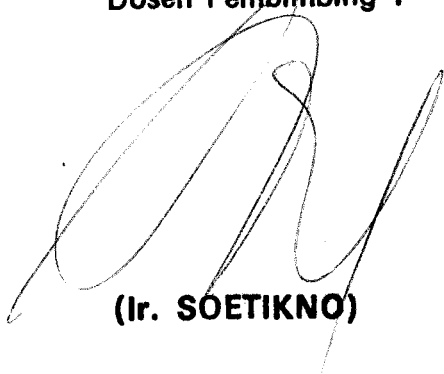
**Fakultas Teknologi Industri**

**Institut Teknologi Sepuluh Nopember**

**S u r a b a y a**

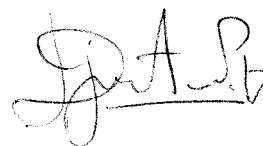
**Mengetahui / Menyetujui**

**Dosen Pembimbing I**



**(Ir. SOETIKNO)**

**Dosen Pembimbing II**



**(Ir. DJOKO PURWANTO)**

**S U R A B A Y A**

***Agustus, 1994***

## ABSTRAK

Perkembangan elektronika membawa dampak positif yang sangat besar dalam menunjang kemajuan dan perkembangan sistem data akuisisi, terutama kehadiran mikroprosesor dalam sistem digital. Salah satu aplikasi dari hal tersebut adalah akuisisi data pada parameter meteorologi.

Dalam tugas akhir ini akan mengaplikasikan suatu alat yang dapat mengakuisisi data sejumlah enam parameter meteorologi, selama selang waktu tertentu.

Dalam pelaksanaan tugas akhir ini alat yang ada dirancang untuk memantau parameter meteorologi berupa suhu udara, tekanan udara, kelembaban udara, kecepatan angin, arah angin, dan curah hujan.

Akuisisi enam parameter diatas dirancang dengan menggunakan mikroprosesor 8088 dan dikomunikasikan secara serial pada komputer IBM-PC.

## KATA PENGANTAR

Penulis memanjatkan puji syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmatnya, sehingga tugas akhir yang berjudul:

### INTERFACE AKUISISI 6 PARAMETER DATA METEOROLOGI DENGAN MINIMUM SISTEM 8088 YANG DIHUBUNGKAN PADA KOMPUTER IBM PC SECARA SERI

Tugas akhir ini merupakan salah satu syarat untuk mendapatkan gelar sarjana pada Jurusan Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, dengan beban 6 SKS (satuan kredit semester). Penulis menyelesaikan tugas akhir ini dengan berdasar pada teori-teori yang didapat dari kuliah, buku-buku referensi, bimbingan dari dosen pembimbing, pengalaman rekan-rekan, serta dorongan dari bapak dan ibu.

Penulis menyampaikan terima kasih yang sebesar-besarnya, kepada:

1. Bapak Ir. Soetikno, selaku dosen pembimbing I, dosen wali, serta koordinator bidang studi yang telah memberikan dorongan dan bimbingan dalam menyelesaikan tugas akhir ini.
2. Bapak Ir. Djoko Purwanto selaku pembimbing II, yang te-

lah memberikan dorongan dan bimbingan dalam menyelesaikan tugas akhir ini.

3. Bapak Dr. Ir. Moch Salehudin, M.Eng.Sc, selaku Ketua Jurusan Teknik Elektro ITS.
4. Rekan mahasiswa jurusan Teknik Elektro khususnya bidang studi Elektronika yang telah mendukung terselesaikannya tugas akhir ini.

Akhirnya penulis berharap tugas akhir ini dapat bermanfaat bagi para pembaca, khususnya peminat dan pihak yang berhubungan dengan akuisisi data meteorologi.

Surabaya, Juli 1994

Penulis

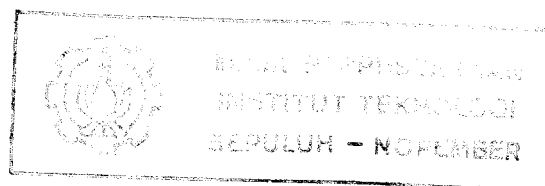
## DAFTAR ISI

BAB	HALAMAN
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xii
 BAB I. PENDAHULUAN.....	 1
I.1. LATAR BELAKANG.....	1
I.2. PERMASALAHAN.....	1
I.3. PEMBATASAN MASALAH.....	2
1.4. TUJUAN.....	3
1.5. METODOLOGI.....	3
1.6. SISTEMATIKA.....	4
1.7. RELEVANSI.....	4
 BAB II. TEORI PENUNJANG.....	 6
II.1. PARAMETER METEOROLOGI.....	6
II.1.1. PARAMETER KECEPATAN ANGIN.....	6
II.1.2. PARAMETER ARAH ANGIN.....	11
II.1.3. PARAMETER KELEMBABAN.....	12
II.1.4. PARAMETER TEMPERATUR.....	17
II.1.5. PARAMETER TEKANAN.....	20

II.1.6. PARAMETER CURAH HUJAN.....	24
II.2. PENGKONDISI SINYAL.....	27
II.2.1. OPERATIONAL AMPLIFIER.....	28
II.3. ADC.....	32
II.4. SISTEM MINIMUM DAN PERANGKAT PHERIPERAL PENDU- KUNGNYA.....	34
II.4.1. MIKROPROSESOR 8088.....	34
II.4.2. PROGAMABLE PHERIPERAL INTERFACE (8255).....	39
II.4.3. PROGAMABLE INTERVAL TIMER (8253).....	43
II.4.4. PROGAMABLE INTERRUPT CONTROLLER (8259).....	48
II.4.5. USART 8251.....	52
 BAB III. PERENCANAAN ALAT.....	61
III.1. PERENCANAAN PERANGKAT KERAS.....	61
III.1.1. PERENCANAAN DIAGRAM BLOK.....	61
III.1.2. PERENCANAAN SENSOR PARAMETER DAN PENG- KONDISI SINYAL.....	65
III.1.3. PERENCANAAN SISTEM MINIMUM 8088.....	83
III.1.4. PERENCANAAN ADC.....	86
III.1.5. PERENCANAAN RANGKAIAN PIT 8253 DAN PIC 8259.....	88
III.1.6. PERENCANAAN USART 8251.....	89
 III.2. PERENCANAAN PERANGKAT LUNAK.....	90
III.2.1. PROGRAM MINIMUM SISTEM.....	90
III.2.2. PETA ALAMAT SISTEM.....	91

III.2.3. PROGRAM PASCAL PADA IBM PC.....	92
BAB IV. PENGUKURAN DAN LINIERISASI.....	96
IV.1. BAGIAN ADC SISTEM.....	96
IV.2. BAGIAN SENSOR KELEMBABAN.....	98
IV.3. BAGIAN SENSOR CURAH HUJAN.....	99
IV.4. BAGIAN SENSOR TEMPERATUR.....	100
IV.5. BAGIAN SENSOR TEKANAN.....	101
IV.6. BAGIAN SENSOR ARAH ANGIN.....	102
IV.7. BAGIAN SENSOR KECEPATAN ANGIN.....	104
BAB V. PENUTUP.....	113
V.1. KESIMPULAN.....	113
DAFTAR PUSTAKA.....	114

#### LAMPIRAN





## DAFTAR GAMBAR

GAMBAR	HALAMAN
2.1. Anemo-Vane.....	7
2.2. Anemometer Portable.....	8
2.3. Anemometer Generator.....	9
2.4. Higrometer Serat.....	13
2.5. Higrometer Resistansi.....	14
2.6. Grafik Kelembaban Relatif Terhadap Resistansi.....	15
2.7. Kapasitansi Sebagai Fungsi Kelembaban Relatif.....	16
2.8. Kerja Dasar Thermocouple.....	17
2.9. Grafik Berbagai Jenis Thermo Couple.....	18
2.10. a. Kurva Karakteristik Jenis RTD.....	19
b. Kurva Karakteristik Jenis Thermistor.....	19
2.11. Jenis Tabung Bourdon.....	21
2.12. Jenis Bellows.....	22
2.13. Bentuk Sensor Tekanan Piezoresistive.....	23
2.14. Jenis Sensor Curah Hujan.....	27
2.15. Inverting Amplifier.....	29
2.16. Non Inverting Amplifier.....	29
2.17. Penguat Differensial.....	30
2.18. Penguat Instrumentasi.....	32
2.19. ADC Jenis Successive Approximation.....	33
2.20. Blok Diagram Sistem Minimum.....	34
2.21. Blok Diagram 8088 Dan Konfigurasi Pin.....	35

2.22. Blok Diagram PPI 8255.....	41
2.23. Format Kontrol Word 8255.....	42
2.24. Blok Diagram PIT 8253.....	44
2.25. Diagram waktu PIT Pada Masing-Masing Mode.....	47
2.26. Blok diagram PIC 8259.....	48
2.27. Format Control Word Dan Diagram Alir Inisialisasi..	50
2.28. Operasi dari command word 8259.....	51
2.29. Format Bit Transmisi Serial Asinkron.....	53
2.30. Blok Diagram Dan Pin 8251.....	54
2.31. Bentuk Mode Word Dan Status Word 8251A.....	58
2.32. Bentuk Command Word 8251A.....	59
2.33. Conector DB-25 dan DB-9.....	60
3.1. Diagram Blok.....	62
3.2. Rangkaian Sensor Kelembaban Dan Astable Multivibrator.....	67
3.3. Penghitung Frekwensi Dengan PIT 8253.....	66
3.4. Bagian Mekanik sensor Kecepatan.....	69
3.5. Rangkaian Sensor Kecepatan Angin.....	70
3.6. Bagian Mekanik Sensor Curah Hujan.....	73
3.7. Rangkaian Sensor Curah Hujan.....	74
3.8. Penghitung Curah Hujan Dengan PIT 8253.....	75
3.9. Rangkaian Sensor Suhu Dengan LM 335.....	78
3.10. Rangkaian sensor tekanan.....	80
3.11. Bagian Mekanik Sensor Arah Angin.....	81
3.12. Bagian Elektronik Sensor Arah Angin.....	82
3.13. CPU Dan Pembangkit Clock.....	83

3.14. Rangkaian Buffer Alamat & Data.....	84
3.15. Rangkaian Memory Sistem Minimum.....	85
3.16. Rangkaian PPI 8255.....	86
3.17. Rangkaian ADC 0804.....	87
3.18. Rangkaian PIT 8253 Dan PIC 8259.....	88
3.19. Rangkaian USART 8251.....	89

## DAFTAR TABEL

TABEL	HALAMAN
2.1. Derajat Kecepatan Angin.....	10
2.2. Arah Angin Dan Kode Gray.....	11
2.3. Perbandingan Sensor Kelembaban.....	16
2.4. Perbandingan Sensor Temperatur.....	20
2.5. Jenis Hujan Yang Terjadi.....	24
2.6. Sistem Decoding Dari Status Bus Cycle.....	39
2.7. Address Pada PPI 8255.....	43
2.8. Address Interval.....	45
2.9. Format Control.....	46
5.1. Bobot Data Pada ADC.....	96
5.2. Pengukuran Kelembaban Terhadap Frekwensi.....	98
5.3. Pengukuran Curah Hujan Terhadap Hitungan Sistem.....	99
5.4. Pengukuran Temperatur Terhadap Bobot ADC.....	100
5.5. Pengukuran Tekanan Terhadap Bobot ADC.....	101
5.6. Pengukuran Sudut Sirip Terhadap Bobot ADC.....	102
5.7. Pengukuran Kecepatan Angin Terhadap Frekwensi.....	104

## BAB I

### PENDAHULUAN

#### I.1. LATAR BELAKANG

Dewasa ini, dunia elektronika telah mengalami perkembangan yang sangat pesat. Hal ini jelas membawa dampak positif yang sangat besar dalam menunjang kemajuan dan perkembangan sistem-sistem elektronika, sistem-sistem komunikasi, sistem-sistem informasi, maupun sistem-sistem instrumentasi.

Pada sistem-sistem instrumentasi khususnya akuisisi data dewasa hampir berperan dalam semua bidang salah satunya adalah bidang meteorologi. Bidang tersebut berperan sangat penting dalam berbagai hal seperti bidang pertanian, bidang penerbangan, bidang pelayaran dan sebagainya. Dari kenyataan tersebut maka perlu suatu alat bantu yang dapat memantau beberapa parameter meteorologi yang dirasa penting untuk menunjang keberlangsungan beberapa bidang diatas.

#### I.2. PERMASALAHAN

Pemantauan beberapa parameter meteorologi seperti suhu udara, tekanan udara, kelembaban udara, kecepatan angin, arah angin dan curah hujan pada saat ini masih dilakukan secara manual, dimana pendataan yang ada masih dilakukan

oleh manusia, dalam interval waktu tertentu dan terus menerus secara kontinyu, hal tersebut tentunya banyak memakan waktu dan tenaga disamping faktor ketelitian manusia.

Untuk mengatasi hal tersebut maka perlu dibuat suatu alat yang dapat memantau secara otomatis dari parameter meteorologi yang ada.

### I.3. PEMBATASAN MASALAH

Untuk memperjelas batasan masalah maka pada tugas akhir ini akan dikhususkan pada beberapa hal:

1. Parameter yang akan diukur sebanyak enam parameter yaitu: suhu udara, tekanan udara, kelembaban udara, kecepatan angin, arah angin, dan curah hujan.
2. Perangkat keras yang dibuat menggunakan sistem minimum 8088 dimana data yang ada dikomunikasikan secara seri pada komputer IBM PC, hal ini dengan maksud agar letak transduser dan mikroprosesor yang ada dapat ditransfer datanya dengan jarak yang relatif jauh.
3. Perangkat lunak yang dibuat dalam dua bentuk yaitu pada sistem minimum 8088 dan pada IBM PC, dalam hal ini data diambil pada interval waktu tertentu, dapat disimpan dalam disket, serta dapat dibuat hasil pengamatan diatas kertas. Dalam hal ini tidak dilakukan analisa hasil pengambilan data.
4. Sensor mekanik yang ada (kecepatan angin, arah angin, curah hujan) didasarkan atas hasil empiris.

#### I.4. TUJUAN

Tujuan dari tugas akhir ini adalah merencanakan dan membuat suatu alat yang mampu mengambil data parameter meteorologi sebanyak enam buah yaitu suhu udara, tekanan udara, kelembaban udara, kecepatan angin, arah angin, dan curah hujan. Pengambilan data tersebut merupakan hasil dari masing-masing keluaran transduser yang dikontrol lebih lanjut pada minimum sistem dan dikomunikasikan pada IBM PC secara serial. Dari keenam transduser yang ada digunakan pengambilan keluaran berupa tegangan dan frekuensi dimana mempunyai hubungan terhadap besarnya parameter yang diukur.

#### I.5. METODOLOGI

Perencanaan alat dalam tugas akhir ini dilakukan dengan pendekatan secara perangkat lunak maupun perangkat keras untuk menghasilkan hasil yang optimum.

Secara perangkat keras, pendekatan dilakukan dengan mempelajari karakteristik transduser, kalibrasi transduser, teknik interfacing dengan minimum sistem dan hal-hal lain yang menunjang seperti pemetaan masukan dan keluaran, pengkodean, penbuferan dan sebagainya.

Secara perangkat lunak pendekatan dilakukan dengan mempelajari perangkat lunak yang memiliki fasilitas dan kemampuan untuk menjalankan sistem tersebut, termasuk linearisasi pada perangkat lunak, pengaturan keluaran dan masukan seperti pengambilan masukan dari keenam transduser,

pengiriman data seri, tampilan pada monitor dan printer.

Selanjutnya dilakukan pengujian alat dengan menguji transduser sebagai masukan dan mengambil output dari hasil proses akuisisi data.

Dan akhirnya disusun buku tugas akhir ini yang mencakup semua langkah dalam perencanaan dan pembuatan tugas akhir.

## **I.6. SISTEMATIKA**

Pada tugas akhir ini disusun dengan sistematika sebagai berikut: Bab I merupakan pendahuluan yang berisi latar belakang, permasalahan, tujuan, metodologi, sistematika, dan relevansi. Bab II berisi tentang teori penunjang dari peralatan elektronika yang berhubungan dengan sistem elektronik, dari sensor yang dipakai sampai sistem minimum beserta periperalnya. Bab III dibahas tentang perencanaan alat dan pembuatan perangkat lunak. Bab IV dibahas mengenai pengukuran dan kalibrasi alat yang dibuat. Bab V berisi kesimpulan dari tugas akhir.

## **I.7. RELEVANSI**

Dengan selesainya tugas akhir ini, diharapkan peralatan akuisisi data dapat diterapkan pada bidang meteorologi sebagai alat bantu untuk pendataan khususnya yang bertalian dengan enam parameter yang ada, sehingga dapat meningkatkan kemampuan dan efektifitas kerja badan meteorologi yang selama ini masih dilakukan secara manual.



Dalam proses pembuatan tugas akhir ini dapat dilihat langkah-langkah merealisasi suatu alat elektronik, dimulai dari analisa alat dan bagian-bagiannya untuk tujuan fungsi tertentu, merencanakannya, merealisasikannya, melakukan uji coba dan mendapatkan ide untuk penyempurnaan lebih lanjut.

## BAB II

### TEORI PENUNJANG

#### II.1. PARAMETER METEOROLOGI

Pada masa sekarang ini meteorologi memegang peranan yang sangat penting, khususnya di bidang pertanian, penerbangan, pelayaran.

Ilmu yang mempelajari tingkah laku udara disebut ilmu meteorologi. Dalam ilmu meteorologi terdapat berbagai macam parameter diantaranya adalah kecepatan angin, arah angin, kelembaban relatif, suhu udara, curah hujan, tekanan udara, pancaran quantum, penguapan, pancaran ultra violet. Khusus dalam tulisan ini akan dibahas parameter meteorologi sebanyak enam jenis yaitu: kecepatan angin, arah angin, suhu udara, tekanan udara, kelembaban udara, curah hujan.

Untuk mengukur besarnya tiap-tiap parameter perlu dibutuhkan alat ukur dari masing-masing parameter tersebut.

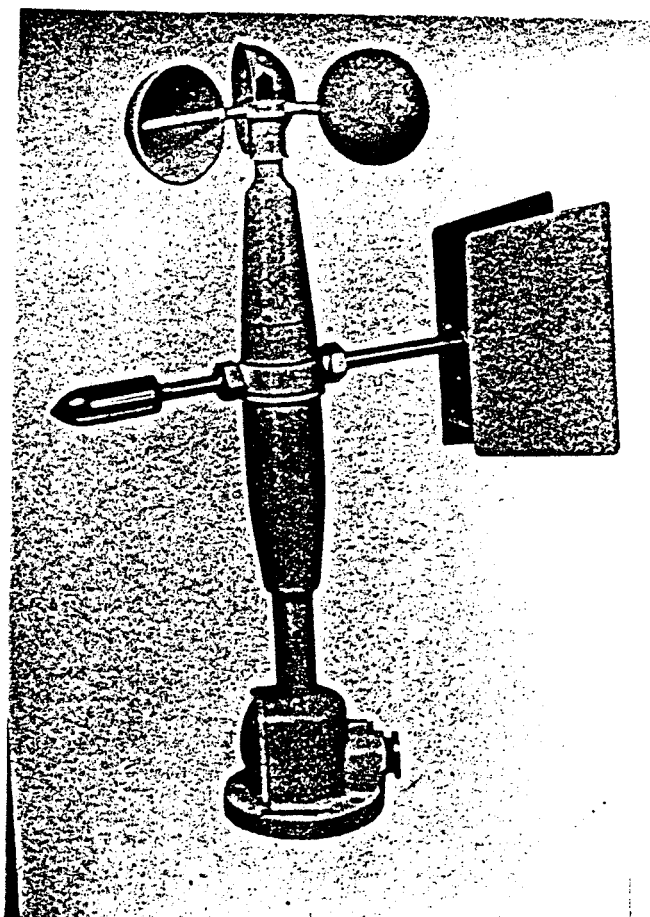
##### II.1.1. PARAMETER KECEPATAN ANGIN

Angin adalah udara yang bergerak, kecepatan angin timbul karena adanya perbedaan tekanan udara, sehingga terjadi gerakan udara dari daerah tekanan tinggi ke daerah tekanan rendah. Kecepatan angin berubah-ubah sesuai ketinggian terhadap permukaan tanah. Untuk mengukur berapa besar kecepatan angin yang terjadi perlu dibutuhkan suatu alat

antara lain:

1. Anemo-vane

Anemo-vane ini digunakan untuk pengukuran secara serempak dari penunjukan kecepatan angin dan arah angin dengan sensitivitas dan ketelitiannya membuat peralatan ini cocok untuk digunakan dalam proyek penelitian mengenai keadaan atmosfer, polusi, bio-meteorologi, pertanian, penerbangan dan lain-lain.



Gambar 2.1 Anemo-vane <sup>1</sup>

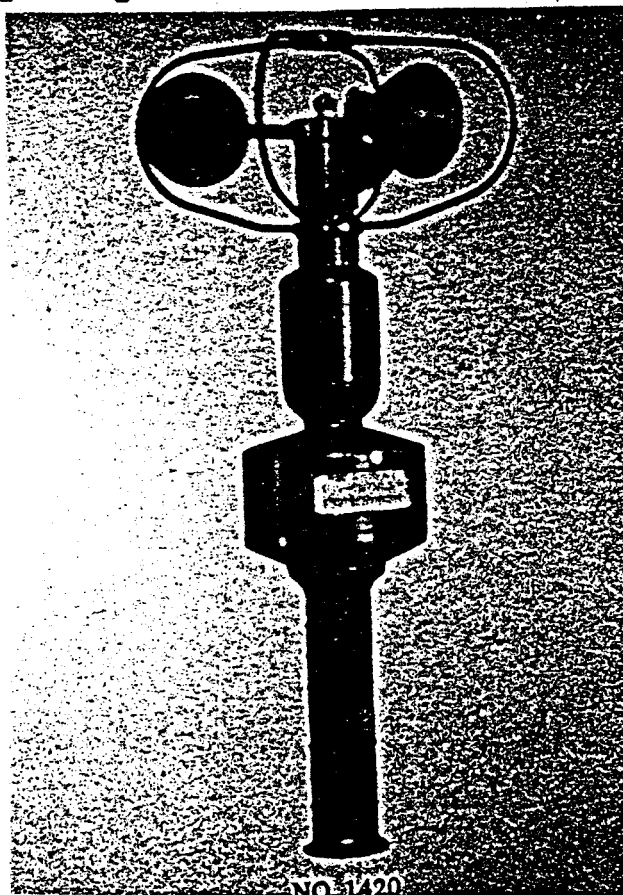
<sup>1</sup>....., Meteorologi Instrument and Hydrographic Instrument, Tamaya & Company Limited, Tokyo, 1978, hal. 20.

Peralatan sensornya terdiri dari sensor kecepatan tiga cup dan sensor arah panah seperti terlihat pada gambar 1

Peralatan penunjukan kecepatan angin dan arah angin dapat terpasang terpisah. Sinyal-sinyal kecepatan angin dan arah angin dibangkitkan didalam sensor unit (lihat gambar 2.1). Peralatan ini mempunyai range pengukuran kecepatan angin 2 meter/detik sampai 60 meter/detik.

## 2. Anemometer Portable

Peralatan ini digunakan untuk mengukur kecepatan angin secara langsung dengan ketelitian 1 meter/detik. Tiga cup



Gambar 2.2 Anemometer Portable<sup>2</sup>

---

<sup>2</sup> Ibid., hal. 19

sensor dipasangkan pada poros rotor, maka bila mendapatkan dorongan angin rotor akan ikut berputar dan menunjukkan skala tertentu sesuai dengan kecepatan angin yang diterima. Untuk sistem ini biasanya digunakan transmisi roda gigi, untuk menskala antara putaran dari poros dengan angka kecepatan pada jarum penunjuk. Lihat gambar 2.2

### 3. Anemometer Generator

Anemometer type generator, merupakan anemometer yang menggunakan sebuah generator untuk merubah putaran menjadi tegangan, dan harga kecepatan angin setara dengan harga GGL yang dibangkitkan



Gambar 2.3 Anemometer generator<sup>3</sup>

<sup>3</sup> Ibid., Hal. 21

### II.1.1.1 DERAJAT KECEPATAN ANGIN

Derajat kecepatan angin biasanya dinyatakan dengan kecepatan dari angin itu berhembus, satuan dari kecepatan angin ini bermacam-macam antara lain m/s, Km/h, mph, knots, skala Beaufort. untuk lebih jelasnya lihat tabel pengelompokan derajat kecepatan angin.

Tabel 2.1 Derajat kecepatan angin <sup>4</sup>

Beaufort	m/s	Diskripsi
0	0 - 0.4	calm
1	0.4 - 1.4	light air
2	1.4 - 3	slight breeze
3	3 - 5	gentle breeze
4	5 - 8	moderate breeze
5	8 - 11	fresh breeze
6	11 - 14	strong breeze
7	14 - 17	high wind
8	17 - 20	gale
9	20 - 24	strong gale
10	24 - 28	whole gale
11	28 - 32	storm
12	32 - 36	hurricane
13	37 - 41	-
14	42 - 46	-
15	47 - 50	-
16	51 - 56	-
17	57 - 60	-

<sup>4</sup> Ruffel, J., "Computer-Controlled Wheather Station," Elektor Elektronik, (January 1992), hal 26.

### II.1.2. PARAMETER ARAH ANGIN

Arah angin menunjukkan arah dari udara mengalir biasanya diterapkan satuan derajat sudut ( $0^{\circ}$ - $360^{\circ}$ ), untuk mengetahui arah angin berhembus biasanya digunakan suatu bentuk sirip. Sistem pengukuran arah angin, antara lain:

#### 1. Sistem kode Gray

Untuk kode Gray sebanyak 4 bit akan didapat resolusi sebesar  $360^{\circ}/16 = 22,5^{\circ}$ . Untuk mendapatkan resolusi yang lebih kecil maka perlu bit yang lebih banyak.

Tabel 2.2 Arah angin dan kode Gray-nya<sup>5</sup>

Arah	Code
N	0000
NNE	0001
NE	0011
ENE	0010
E	0110
ESE	0111
SE	0101
SSE	0100
S	1100
SSW	1101
SW	1111
WSW	1110
W	1010
WNW	1011
NW	1001
NNW	1000

<sup>5</sup> Ibid., hal. 29

## 2. Sistem Potensiometer

Pada sistem ini putaran dari sirip, menghasilkan posisi yang memutar wiper dari potensiometer yang telah diberi tegangan exitasi tertentu, sehingga keluaran tegangannya merepresentasikan besarnya posisi dari sirip arah angin tersebut. Untuk sistem ini resolusi yang didapatkan dapat mencapai  $1^\circ$ . Pada sistem ini perlu dibutuhkan potensiometer khusus yang dapat berputar sebesar  $360^\circ$ .

### II.1.3. PARAMETER KELEMBABAN

Kelembaban merupakan parameter yang berhubungan dengan kandungan uap air dalam udara, berapa banyak jumlah uap air yang dapat dikandung oleh udara tergantung beberapa faktor, faktor yang paling penting adalah temperatur udara, udara hangat mengandung uap air lebih banyak dibanding udara dingin. Dari definisi yang ada pengukuran kelembaban dibagi menjadi beberapa macam antara lain:

1. Kelembaban absolut: kandungan uap air yang terdapat dalam sejumlah udara tertentu. Satuan dari kelembaban absolut adalah satuan massa/satuan volume ( $\rho$ ).

2. Kelembaban relatif: perbandingan antara kerapatan (density) uap air pada saat itu ( $\rho_h$ ) terhadap kerapatan air pada saat jenuh ( $\rho_s$ ), dimana bila kita aplikasikan dalam bentuk persamaan matematikanya adalah:

$$RH = ( \rho_h / \rho_s ) \times 100\% \dots \dots \dots \text{dalam } (\%)$$



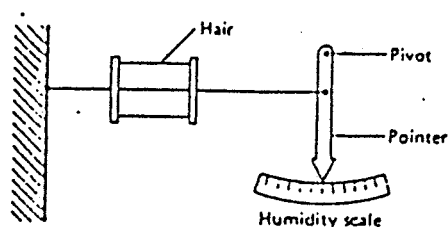
Untuk mengetahui kelembaban dari udara perlu dibutuhkan sensor kelembaban, untuk sistem ini besaran yang diukur adalah kelembaban relatif. Sensor yang sering digunakan untuk mengukur kelembaban relatif antara lain adalah:

### 1. Psychrometer

Merupakan penentu kelembaban dengan sistem pembacaan dua termometer, satu termometer bola kering (membaca termometer udara biasa) sedangkan yang lain adalah termometer bola basah, digunakan untuk menentukan temperatur jenuh, Dari perbedaan pembacaan kedua temperatur tersebut maka dapat ditentukan kelembaban yang terukur (melalui tabel psychrometer).

### 2. Higrometer Serat

Merupakan higrometer yang paling sederhana, karena elemennya berasal dari rambut manusia atau kulit binatang. Rambut manusia akan bertambah panjang 3% pada range kelembaban 0%-100%, perubahan ini akan terbaca bila serat tersebut dihubungkan dengan jarum penunjuk, lihat gambar 2.4.

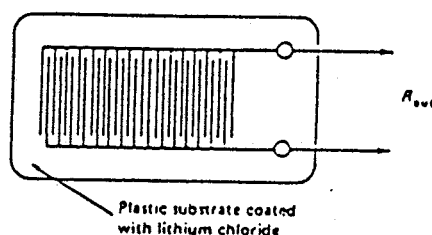


Gambar 2.4 Higrometer Serat<sup>6</sup>

<sup>6</sup> Schuler, Charles A., McNamee, William L., Industrial Electronic And Robotic, McGrawhill, Inc, Singapore, 1986, hal. 165

### 3. Higrometer Resistansi

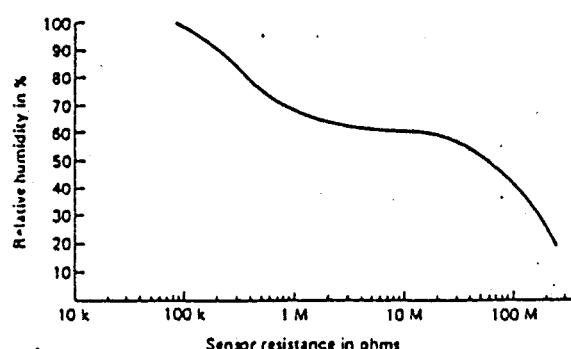
Merupakan higrometer yang bekerjanya atas dasar prinsip resistansi karena perubahan kelembaban. Higrometer resistansi sebagaimana terlihat pada gambar 2.5, tersusun atas dua buah elektroda yang saling berhadapan dan dipisahkan oleh lapisan tipis film lithium-chlorida. Film lithium-chlorida bersifat higroskopis, artinya mampu menyerap uap air dari udara.



Gambar 2.5 Higrometer Resistansi<sup>7</sup>

Perubahan yang terjadi adalah saat kelembaban relatif 100% resistansi sensor tersebut 80 K dan pada saat kelembaban relatif 20% resistansi sensor tersebut menjadi 300 M. Untuk jelasnya lihat grafik gambar 2.34. Dalam rangkaian biasanya dirangkai dalam bentuk jembatan, dengan sumber exitasi AC, sebab bila dirangkai dengan DC maka film lithi-um-clorida akan terpolarisasi dan rusak.

<sup>7</sup> Ibid., hal 219.



Gambar 2.6. Grafik kelembaban relatif terhadap resistansi<sup>8</sup>

#### 4. Higrometer Kapasitansi

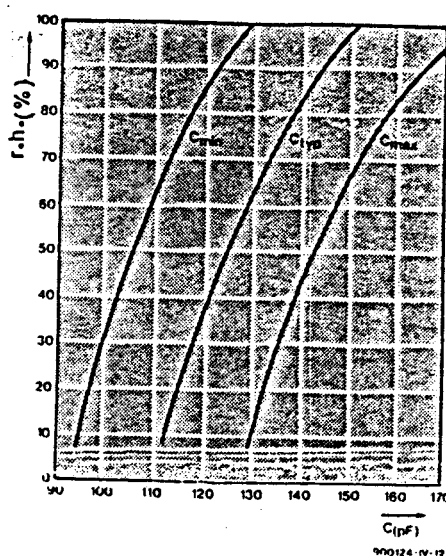
Merupakan higrometer yang prinsip kerjanya karena adanya perubahan konstanta dielektrik, dengan berubahnya konstanta dielektrik maka akan berubah pula besarnya kapasitansi yang ada, sesuai dengan rumus :

$$C = (\epsilon_r \times A) / D \dots \dots \dots \text{Farad}$$

Dimana  $\epsilon_r$  merupakan konstanta dielektrik dari kondensator,  $A$  merupakan penampang luas kondensator,  $D$  merupakan jarak keping dari kondensator. Terlihat dengan adanya perubahan  $\epsilon_r$  maka akan berubah pula besarnya kapasitansi yang ada, untuk lebih mudah mengadakan pengukuran biasanya digunakan pengkonversi dari kapasitansi menjadi tegangan. Dalam tugas

<sup>8</sup>Ibid., Hal 165

akhir ini digunakan higrometer kapasitansi, dalam gambar 1.7 diperlihatkan karakteristik higrometer kapasitansi buatan Philips.



Gambar 2.7 Kapasitansi sebagai fungsi kelembaban relatif<sup>9</sup>

Tabel 2.3 Perbandingan sensor kelembaban

Name	Operating Principle	Relative Humidity Range	Temperature Range	Accuracy
Psychrometer	Evaporation rate of water	2%-98%	32°-212°F	± 5%
Hair hygrometer	Hair changes length with humidity	15%-90%	0°-160°F	± 5%
Resistance hygrometer	Lithium chloride changes resistance with humidity	1.5%-99%	-40°-160°F	± 1.5%
Resistance-capacitance hygrometer	Aluminum oxide changes impedance with humidity	0%-100%	0°-300°F	± 2%
Capacitive hygrometer	Dielectric constant changes with humidity	0%-100%	-200°-200°F	± 0.5%

<sup>9</sup> Rietjens, A., "Digital Hygrometer", Elektor Electronics, (November 1993), hal. 18.

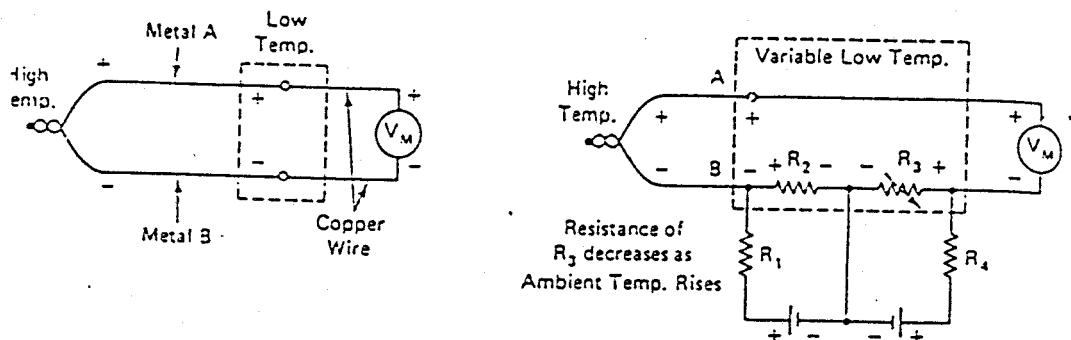


#### II.1.4. PARAMETER TEMPERATUR

Parameter ini berhubungan dengan besarnya suhu udara diukur. Dewasa ini terdapat berbagai macam bentuk sensor untuk mengukur temperatur diantaranya adalah:

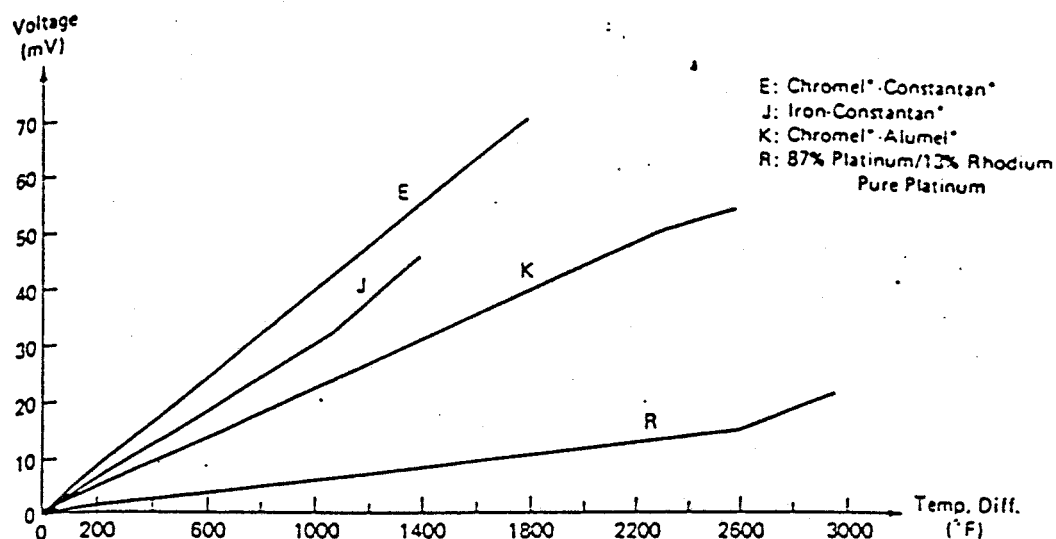
##### 1. Thermocouple

Thermocouple merupakan sensor yang sering digunakan dalam dunia industri untuk mengukur temperatur. Thermocouple terdiri dari sepasang kawat metal dengan bahan tertentu yang berbeda yang ujungnya dihubungkan dalam satu titik. Bila ujung dari thermocouple diberikan suatu suhu tertentu maka pada thermocouple akan terbangkit perbedaan tegangan yang tertentu pula (Seebeck effect). Thermocouple terdiri dari beberapa macam jenis diantaranya adalah thermocouple jenis E, jenis J, jenis K, jenis R



Gambar 2.8 Kerja dasar dari thermocouple<sup>10</sup>

<sup>10</sup> Maloney, Timothy J., Industrial Solid State Electronics Devices and System, Prentice Hall, Inc., Englewood Cliffs, New Jersey 0763, Hal. 408

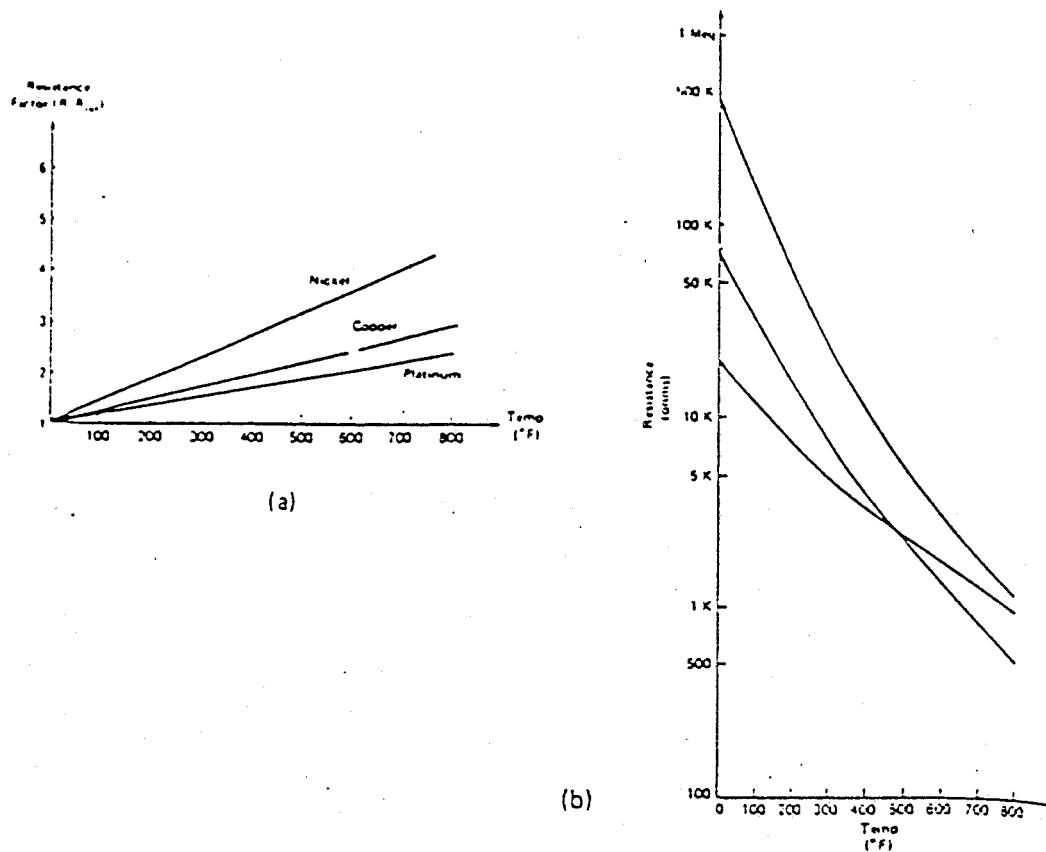


Gambar 2.9 Grafik dari berbagai jenis thermocouple<sup>11</sup>

## 2. Thermistor dan Resistive Temperature Detectors(RTD)

Sensor ini bekerja atas dasar berubahnya resistansi suatu bahan karena adanya pengaruh temperatur. Bahan yang dipakai biasanya merupakan metal murni (untuk RTD) dan metal oksida (untuk thermistor), kedua jenis bahan ini mempunyai koefisien temperatur yang berbeda, pada metal murni mempunyai koefisien temperatur positif, sedangkan pada metal oksida mempunyai koefisien temperatur negatif. Untuk mendapatkan informasi temperatur yang terukur pada sensor jenis ini diperlukan sumber exitasi dari luar.

<sup>11</sup> Ibid., Hal. 408



Gambar 2.10<sup>12</sup>a) Kurva karakteristik berbagai jenis RTD.

b) Kurva karakteristik berbagai jenis Thermistor





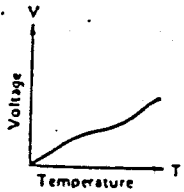
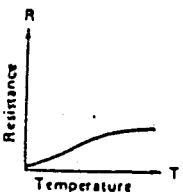
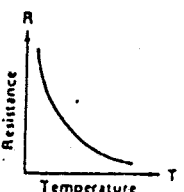
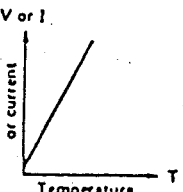
### 3. Sensor IC Temperatur

Pada IC sensor mempunyai koefisien temperatur tertentu, Koefisien tersebut berupa tegangan terhadap suhu atau arus terhadap suhu. Untuk koefisien tegangan terhadap suhu adalah LM 135, LM 235, LM 335, AD 590 sedangkan untuk koefisien arus terhadap suhu adalah LM 134, LM 234, LM 334 untuk lebih jelasnya lihat data book pada lampiran. Dari beberapa jenis

<sup>12</sup> Ibid., Hal. 412

sensor yang dijelaskan, dalam tugas akhir ini dipilih sensor dalam bentuk IC (lihat tabel perbandingan).

Tabel 2.4 Perbandingan sensor Temperatur<sup>13</sup>

	Thermocouple	RTD	Thermistor	IC Sensor
				
				
Advantages	<input type="checkbox"/> Self-powered <input type="checkbox"/> Simple <input type="checkbox"/> Rugged <input type="checkbox"/> Inexpensive <input type="checkbox"/> Wide variety <input type="checkbox"/> Wide temperature range	<input type="checkbox"/> Most stable <input type="checkbox"/> Most accurate <input type="checkbox"/> More linear than thermocouple	<input type="checkbox"/> High output <input type="checkbox"/> Fast <input type="checkbox"/> Two-wire ohms measurement	<input type="checkbox"/> Most linear <input type="checkbox"/> Highest output <input type="checkbox"/> Inexpensive
Disadvantages	<input type="checkbox"/> Non-linear <input type="checkbox"/> Low voltage <input type="checkbox"/> Reference required <input type="checkbox"/> Least stable <input type="checkbox"/> Least sensitive	<input type="checkbox"/> Expensive <input type="checkbox"/> Power supply required <input type="checkbox"/> Small $\Delta R$ <input type="checkbox"/> Low absolute resistance <input type="checkbox"/> Self-heating	<input type="checkbox"/> Non-linear <input type="checkbox"/> Limited temperature range <input type="checkbox"/> Fragile <input type="checkbox"/> Power supply required <input type="checkbox"/> Self-heating	<input type="checkbox"/> $T < 200^\circ\text{C}$ <input type="checkbox"/> Power supply required <input type="checkbox"/> Slow <input type="checkbox"/> Self-heating <input type="checkbox"/> Limited configurations

### II.1.5. PARAMETER TEKANAN

Parameter ini berhubungan dengan tekanan udara, dewasa ini terdapat bermacam-macam sensor tekanan antara lain:

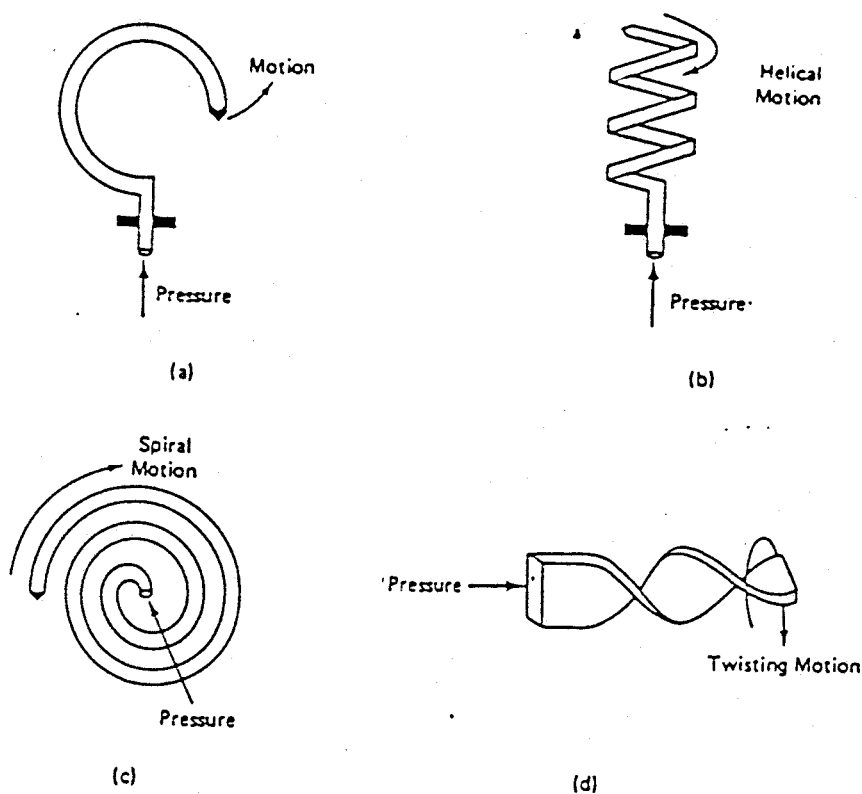
#### 1. Tabung Bourdon

Tabung Bourdon dibentuk dari bahan metal dengan penampang melintang berbentuk oval, pada bagian ujung tertutup dan pada bagian pangkal terbuka. Bila terdapat tekanan

<sup>13</sup> Charles A., Schuler, William L, McNamee, Op. cit., hal. 214



tertentu pada bagian pangkal tabung maka tabung akan merenggang proporsional dengan tekanan yang masuk. Tabung bourdon terdapat beberapa jenis antara lain tabung bentuk helix, spiral, bentuk C (biasanya dikombinasikan dengan sebuah LVDT atau elemen resistansi). Pada jenis ini dapat mengukur tekanan dari 10 psi sampai 300 psi.

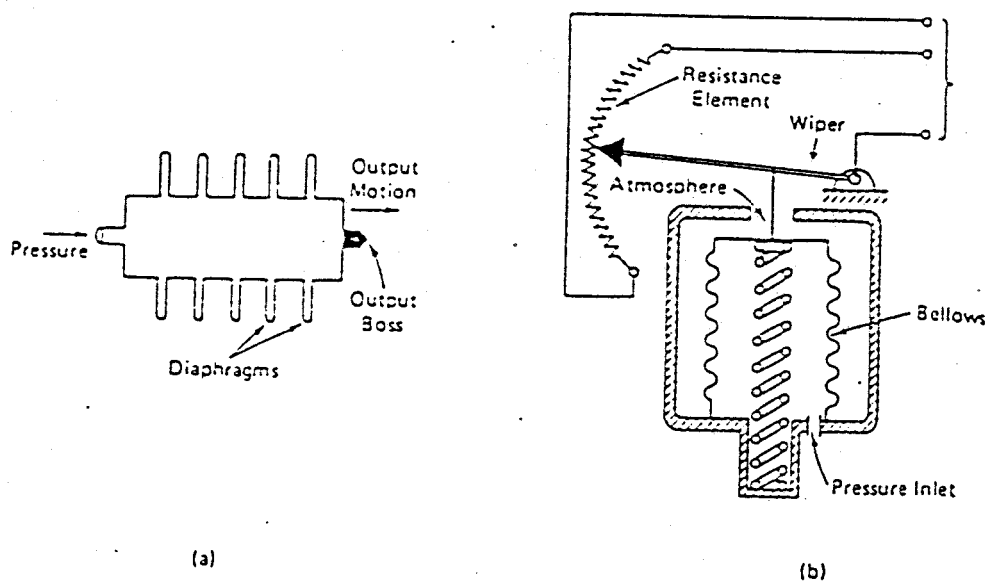


Gambar 2.11 Jenis tabung Bourdon<sup>14</sup>

<sup>14</sup> Opcit., hal.405

## 2. Bellows

Bellows merupakan sensor tekanan yang terdiri dari diafragma seri dengan bahan metal elastis. Bila tekanan terjadi pada bagian yang terbuka maka bellows akan mengembang porposional dengan besarnya tekanan yang terjadi. Bellows biasanya dikombinasikan dengan sebuah LVDT atau elemen resistansi. Pada jenis ini dapat mengukur tekanan dari 0.5 psi sampai 20 psi.



Gambar 2.12 Jenis bellows<sup>15</sup>

<sup>15</sup> Ibid., hal. 406

### 3. Piezoresistive

Pada sensor tekanan jenis ini memanfaatkan prinsip dasar dari sifat piezoresistive dalam silikon. Adanya tekanan pada difragma pada sensor akan menyebabkan perubahan resistivitas yang proporsional terhadap tekanan yang terjadi. Dari transduser ini dapat mengukur tekanan sesuai dengan type komponen (0-7,5 psi, 0-15 psi, 0-30 psi), pabrik yang memproduksi sensor ini antara lain adalah Sensym, Motorola.

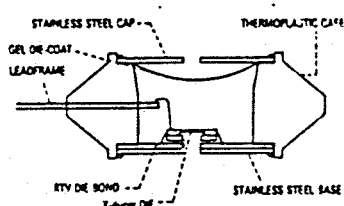
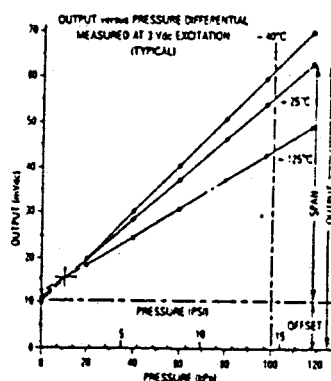


Figure 1. Motorola X-ducer Pressure Sensor Package



Gambar 2.13 Bentuk sensor tekanan piezoresistive<sup>16</sup>

<sup>16</sup> Hogenboom, P., Data Sheet Book 3, Elektuur B.V., Beek L, Netherlands, hal. 324

### II.1.6. PARAMETER CURAH HUJAN

Peristiwa jatuhnya tetes-tetes air dari langit disebut hujan. Hujan sering diukur dalam berapa besar curah hujannya atau berapa besar intensitas curah hujan. Satuan yang dipakai untuk mengukur curah hujan adalah mm sedangkan untuk mengukur intensitas curah hujan adalah mm/jam.

Curah hujan =  $dV/A$ .....dalam mm

dimana  $dv$  = jumlah pertambahan volume air hujan ( $ml^3$ ).

$A$  = luas penampang sensor( $mm^2$ ).

Hujan yang terjadi di alam ini mempunyai ukuran dari pada butir-butir yang berjenis-jenis, adapun jenis hujan tergantung dari ukurannya. Dalam meteorologi, butir hujan dengan diameter lebih besar dari 0.5 mm disebut hujan sedangkan yang berdiameter antara 0.1 - 0.5 disebut gerimis, makin besar butir hujan yang terjadi makin besar pula kecepatan jatuhnya. Tabel dibawah ini menunjukkan jenis hujan yang terjadi.

Tabel 2.5 Jenis hujan yang terjadi<sup>17</sup>

Jenis	Diameter (mm)	Kec. Jatuh(m/s)
Hujan Gerimis	0.15	0.5

<sup>17</sup> Penelitian Air Bersih, Dinas Kesehatan RI, 1979, Hal. 16

Hujan halus	0.5	2.1
Hujan normal		
-Lemah	1.0	4.0
-Deras	2.0	6.5
Hujan sangat		
deras	3.0	8.1

Untuk mengukur curah hujan perlu digunakan sensor curah hujan, yang digunakan sebagai penakar curah hujan, ada beberapa macam jenis sensor curah hujan.

#### 1. Sensor Manual

Sensor ini biasanya digunakan pada jawatan meteorologi, dimana penakar curah hujan berbentuk tabung dengan diameter tertentu yang diletakan pada areal lahan, dan pengukurannya cukup dilakukan dengan gelas ukur, dari volume yang didapat maka dapat ditentukan curah hujanya.

#### 2. Sensor Model Tetesan

Sensor ini menghitung curah hujan dengan prinsip menghitung jumlah tetesan air pada tabung penampung. kederaan hujan sangat mempengaruhi kecepatan jatuhnya tetesan air melalui pipa sensor, tetesan yang terjadi jatuh melalui pipa sensor dan menghalangi lampu yang menuju ke photo dioda, hal ini akan menghasilkan sebuah sinyal untuk setiap tetes air, untuk jenis ini dirasa kurang presisi karena besar tetesan

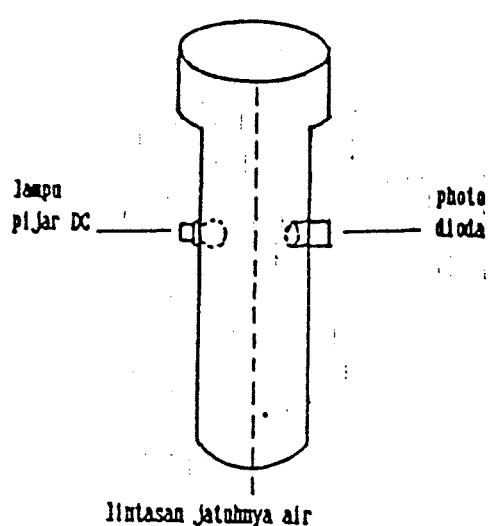
air tidak selalu homogen.(lihat gambar)

### 3. Sensor Model Bucket (timbangan)

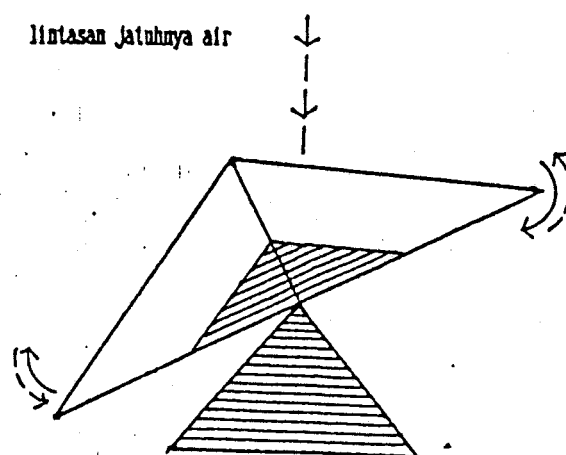
Sensor ini menghitung besarnya curah hujan dengan jalan menggunakan timbangan seperti pada gambar. Air hujan yang jatuh dari langit ditampung dalam tabung dan langsung dikeluarkan melalui saluran kecil di bagian bawah tabung pada saat itu juga. Air mengalir dari saluran tersebut ditampung dalam satu sisi pada timbangan, dan bila satu sisi sudah penuh maka timbangan akan bergulir dengan sendirinya dan air yang mengalir akan ditampung pada sisi timbangan yang lain, gerakan naik turun tersebut akan memotong saklar magnetik dan akan menghasilkan hitungan. Kederasan air hujan sangat berpengaruh pada mengalirnya air yang melalui pipa kecil., hal ini tidak banyak berpengaruh pada alat ukur ini karena tiap pulsa yang dihasilkan sudah mencerminkan jumlah volume tertentu (dari timbangan yang ada).

Sensor ini mempunyai prinsip yang sama dengan model tetesan yaitu merubah besaran volume air menjadi besaran digital, bedanya pada sensor model ini besaran volumenya sudah pasti dan selalu konstan (tidak dipengaruhi oleh besarnya butir-butir tetesan). Dari uraian tersebut maka alat ukur curah hujan untuk model ini dirasa lebih baik dibanding dengan model tetesan, karena tiap pulsa digital yang terjadi selalu sebanding dengan jumlah volume air dalam timbangan. Pada tugas akhir ini digunakan model timbangan, dengan sensor gerakannya digunakan saklar magnetic.

# TABUNG PENAMPUNG AIR HUJAN



-Model sensor Tetesan



-Model sensor Timbangan

Gambar 2.14 Jenis sensor curah hujan

## II.2 PENGKONDISISI SINYAL

Sebelum sinyal dari masing-masing sensor dapat diolah oleh mikrokomputer maka besaran-besaran listrik dari keluaran sensor harus dikondisikan sinyalnya terlebih dahulu, sehingga besaran tersebut memenuhi syarat untuk diolah dalam mikrokomputer.

### II.2.1. OPERASIONAL AMPLIFIER

Dalam rangkaian sistem didapat sinyal listrik dari sensor yang masih terlalu lemah untuk diproses lebih lanjut, maka banyak pengkondisi sinyal yang berbentuk amplifier, khususnya amplifier yang menggunakan operasional amplifier. Untuk memudahkan analisa maka op-amp diasumsikan sebagai opamp ideal yang mempunyai sifat sebagai berikut:

1. Penguatan terbuka tak berhingga ( $A_{ol} = \infty$ ).
2. Impedansi masukan tak berhingga ( $Z_{in} = \infty$ ).
3. Impedansi output nol ( $Z_o = 0$ ).
4. Lebar pita tak berhingga ( $BW = \infty$ ).
5. Differensial input nol ( $E_d = 0$ ).

Op-amp dapat difungsikan menjadi fungsi tertentu diantaranya:

#### 1. Op-amp sebagai penguat Inverting<sup>18</sup>

Inverting amplifier adalah suatu amplifier yang mempunyai penguatan negatif, artinya polaritas sinyal input berlawanan dengan polaritas sinyal output. Rangkaian dasarnya dapat dilihat pada gambar.

Penurunan rumus:

$$i_{Ri} = i_{Rf}$$

$$V_i/R_i = (0 - V_o)/R_f$$

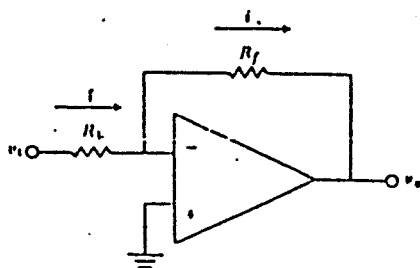
$$V_o/V_i = -R_f/R_i$$

---

<sup>18</sup> Coughlin, Robert F., Driscoll, Fedrick F., Penguat Operasional dan Rangkaian Terpadu Linier, Erlangga, Jakarta, 1985, hal. 33



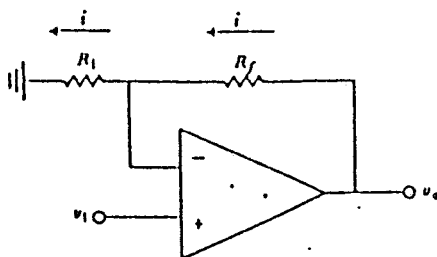
Jadi  $A_{cl} = -R_f/R_i$



Gambar 2.15 Inverting Amplifier

## 2. Op-amp sebagai penguat Non Inverting<sup>19</sup>

Inverting amplifier adalah suatu amplifier yang tak membalikan polaritas sinyal hasil penguatan (lihat gambar).



Gambar 2.16 Non inverting amplifier

<sup>19</sup> Ibid., hal 45

Penurunan rumus:

$$i_{Ri} = i_{Rf}$$

$$(0 - V_i)/R_i = (V_i - V_o)/R_f$$

$$-R_f.V_i = R_i.V_i - R_i.V_o$$

$$R_i.V_o = (R_i + R_f).V_i$$

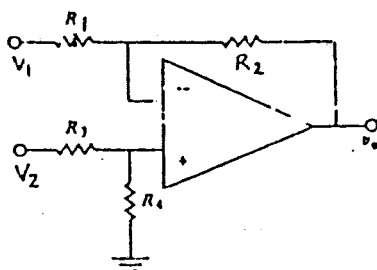
$$V_o/V_i = (R_i + R_f)/R_i$$

$$V_o/V_i = 1 + R_f/R_i$$

$$A_{cl} = 1 + R_f/R_i$$

### 3. Op-amp sebagai penguat Differensial<sup>20</sup>

Differensial amplifier adalah merupakan gabungan dari non inverting dan inverting amplifier. Rangkaian terlihat pada gambar.



Gambar 2.17 Penguat Differensial

Penurunan rumus:

Bila diterapkan prinsip superposisi maka berlaku:

<sup>20</sup>Ibid., hal 161

$$V_o = V_{o1} + V_{o2}$$

Untuk inverting amplifier

$$V_{o1} = R_2/R_1 \cdot V_1$$

Untuk non inverting amplifier

$$\begin{aligned} V_{o2} &= (R_1 + R_2)/R_1 \cdot R_4/(R_3 + R_4) \cdot V_2 \\ &= (R_2 + R_1)/(R_3 + R_4) \cdot R_4/R_1 \cdot V_2 \end{aligned}$$

Jadi:

$$V_o = -R_2/R_1 \cdot V_1 + [(R_2 + R_1)/(R_3 + R_4) \cdot R_4/R_1 \cdot V_2]$$

Jika  $R_1 = R_2$

maka:

$$\begin{aligned} V_o &= -R_2/R_1 \cdot V_1 + [R_2/R_1 \cdot V_2] \\ &= (V_2 - V_1) \cdot R_2/R_1, \text{ jika } V_2 - V_1 = V_d, \text{ maka:} \end{aligned}$$

$$V_o = R_2/R_1 \cdot V_d$$

#### 4. Op-amp sebagai Voltage Follower<sup>21</sup>

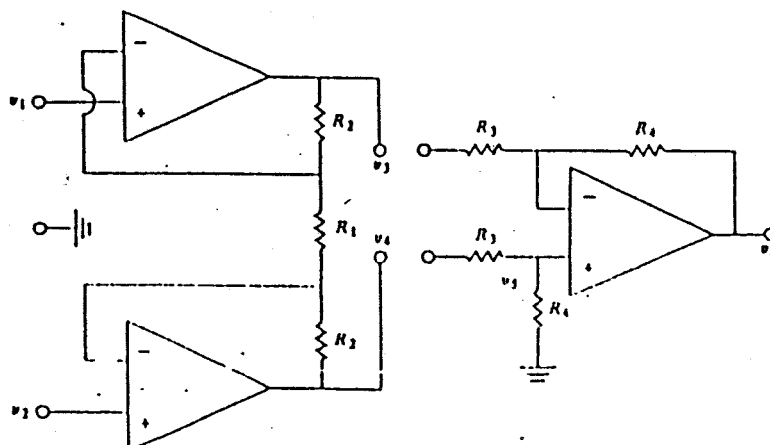
Op-amp sebagai voltage follower merupakan penguat non inverting yang mempunyai besar penguatan sebesar satu kali. Ditinjau dari sifat op-amp yang mempunyai impedansi input yang besar dan impedansi output yang kecil, maka op-amp yang dioperasikan pada Voltage follower sering digunakan sebagai penyangga yang membutuhkan suatu sifat matching impedance, dengan adanya penyangga tersebut maka diharapkan akan terjadi transfer beban maximum tanpa ada kerugian karena efek pembebanan.

---

<sup>21</sup> Ibid., hal. 42

### 5. Op-amp sebagai Penguat Instrumentasi<sup>22</sup>

Penguat Instrumentasi merupakan perpaduan antara op-amp sebagai Follower dan op-amp sebagai penguat Differensial, lihat gambar:



Gambar 2.18 Penguat Instrumentasi

### II.3 ADC (ANALOG TO DIGITAL CONVERSION)

ADC adalah perangkat yang dipakai untuk mengubah sinyal listrik analog menjadi sinyal digital. ADC mempunyai jenis yang berbeda-beda sesuai dengan prinsip kerjanya, antara lain: successive approximation, integrasi, pencacah, flash. Dalam sistem yang dibuat digunakan sistem jenis successive

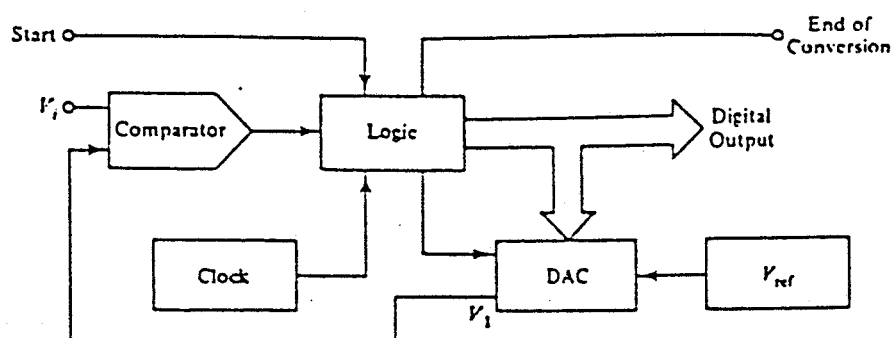
<sup>22</sup> Gayakwad, Ramakant A., *Op-amp and Linier Integrated Circuit*, 2nd, Prentice Hall of India, New Delhi, 1988, Hal. 236

aproximation karena mempunyai kelebihan - kelebihan antara lain waktu konversi cukup cepat.

ADC jenis succesive approximation menggunakan register pendekatan berurutan untuk merubah besaran analog menjadi digital. Urutan kerjanya sebagai berikut:

- MSB diset ke logika 1 dan bit-bit lainnya diisi dengan 0.

Hal ini menghasilkan harga  $V_{out}$  pada keluaran DAC sama dengan bobot dari MSB, apabila  $V_{out}$  lebih besar dari  $V_{in}$ , maka keluaran pembanding akan berubah ketaraf rendah, hal ini akan mereset MSB, apabila  $V_{out}$  lebih kecil dari  $V_{in}$ , MSB tetap pada logika 1.



Gambar 2.19 ADC jenis succesive approximation<sup>23</sup>

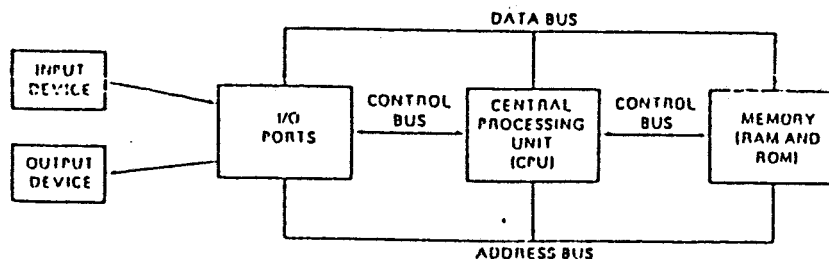
<sup>23</sup>Schuler, charles A.,McNamee, William L., Op.cit.,hal. 358

-MSB kedua diset ke 1, apabila harga baru dari  $V_{out}$  lebih besar  $V_{in}$ , bit ini akan direset ke 0, bila lebih kecil harganya tetap 1.

-Proses ini berlangsung terus untuk seluruh bit dalam register. Proses pendekatan ini memerlukan satu perioda clock untuk tiap bit, hal ini merupakan kelebihan dari SA-ADC karena dengan 8 clock dapat menyelesaikan setiap konversi (untuk ADC 8 bit). Untuk lebih jelas lihat lampiran data.

#### II.4. SISTEM MINIMUM 8088 DAN PERIPHERAL PENDUKUNGNYA

Untuk mengkoordinasi sistem yang dibuat digunakan mikriprosesor 8088 yang dioperasikan pada mode minimum. Sesuai dengan keperluan maka sistem ini didukung beberapa peripheral seperti PPI 8255, PIT 8253, PIC 8259, USART 8251.

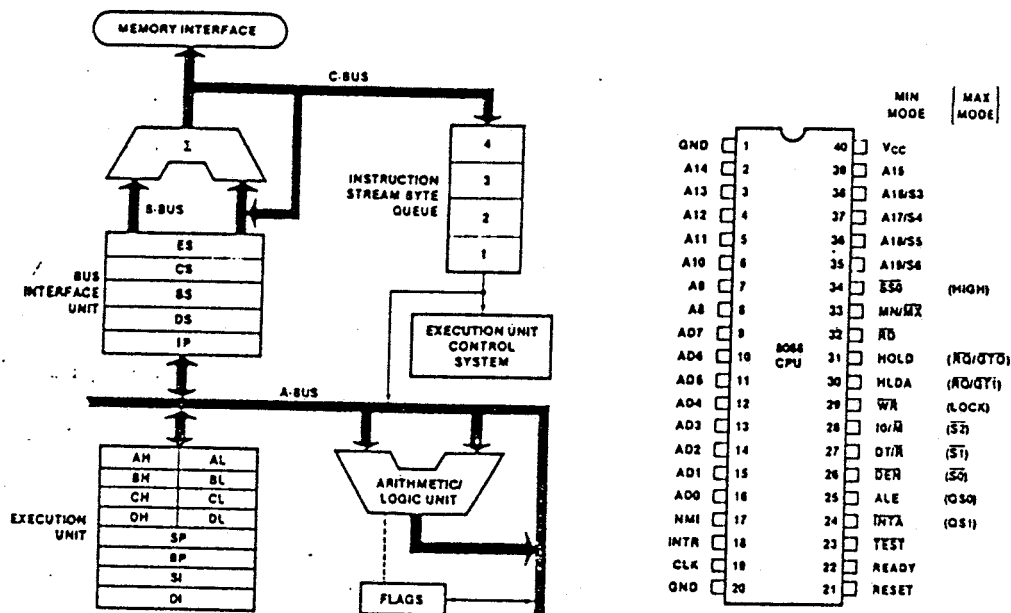


Gambar 2.20 Blok diagram sistem minimum

##### II.4.1. MIKROPROSESOR 8088

Mikroprosessor 8088 tersusun atas dua bagian yaitu

1. Bus interface unit (BIU)
2. Execution Unit (EU)



Gambar 2.21 Blok diagram dan konfigurasi pin

Bus interface unit mempunyai fungsi diantaranya membangkitkan alamat memory atau I/O untuk transfer data antara peralatan periperal dan 8088, execution unit menerima kode instruksi program dan data dari BIU, melaksanakan instruksi tersebut, memanipulasi data, menyimpan data dalam register dan BIU mengambilnya untuk disimpan dalam memory atau diki-

rim ke I/O. Berikut penjelasan pin-pin dari pada mikroprosesor 8088 yang digunakan dalam mode minimum<sup>24</sup>

-AD0-AD7

Merupakan jalur address dan data yang bekerja secara multipleks. Jalur ini bekerja sebagai address(A0-A7) bus cycle pertama (t1) dan pada bus cycle berikutnya (t2-t4) jalur ini berfungsi sebagai jalur data.

-A8-A15

Merupakan address bus untuk seluruh bus cycle (t1-t4), bus ini tidak memerlukan latch oleh ALE, bus ini akan float/tristate off selama interrupt acknowledge atau local bus acknowledge.

-A16/S3 -A19/S6(Address/Status)

Jalur ini berfungsi sebagai jalur address dan status yang bekerja secara multipleks. Selama bus cycle yang pertama (T1), jalur ini berfungsi sebagai address (A16-A19) untuk operasi memory, selama operasi I/O jalur ini akan low. Untuk bus cycle berikutnya jalur ini berfungsi sebagai penunjuk status dari 8088.

-RD (Read)

RD merupakan Read Strobe, bilamana aktif berarti mikroprosesor sedang melaksanakan pembacaan isi memory atau melaksanakan pembacaan dari port. Jika IO/M berkeadaan 0 berarti se-

---

<sup>24</sup>Intel Corporation, Microkomputer And Periperal Hand Book Vol I, Intel Corporation Literature Distribution, Santa clara, 1987, Hal. 2-90



dang membaca memory, tetapi jika 1 berarti sedang membaca port.

#### -READY

Input untuk sinyal dari memory atau I/O yang menunjukkan transfer data selesai, sinyal dari memory atau dari I/O ini disinkronkan dulu oleh clock generator sebelum masuk ke pin Ready.

#### -INTR

Sinyal ini merupakan permintaan interupsi ke mikroprosesor 8088 dari perangkat atau komponen luar.

#### -TEST

Input ini menentukan kerja prosesor jika ada intruksi yang bersifat menunggu untuk suatu uji, jika input ini 0 maka esekusi akan dilanjutkan, jika 1 maka prosesor akan menunggu.

#### -NMI (non maskable interrupt)

Input untuk menginterrupt 8088, interrupt ini merupakan interrupt tipe 2, interrupt ini merupakan interrupt yang tidak bisa dihalangi.

#### -RESET

Dengan adanya sinyal reset ini, maka mikrokomputer 8088 akan menjalankan instruksi mulai dari awal lagi. Sinyal input harus dipertahankan tinggi selama paling sedikit 4 siklus clock.

#### -CLK

CLK merupakan sinyal input clock untuk 8088 dan bus control-

er dan berfungsi untuk mengatur timing dari operasi 8088 dan bus controller. Bentuk pulsa clock harus memiliki rise time 10ns dan duty cycle 33%.

-VCC

Merupakan pin power supply, besar tegangan +5 V.

-MN/-MX (minimum/maximum)

Merupakan pin yang menyatakan mode operasi dari 8088. Jika input pin ini berlogika 1 maka 8088 beroperasi dengan mode minimum.

-IO/M

Berfungsi untuk membedakan antara memory access atau I/O access.

-WR (Write)

Signal ini menyatakan bahwa mikroprosesor 8088 sedang melaksanakan operasi write memory atau I/O, tergantung dari pin IO/M.

-INTA (interrupt Acknowledge)

Menyatakan bahwa mikroprosesor 8088 telah mendeteksi adanya permintaan interrupt.

-ALE (address latch enable)

Merupakan signal out dari 8088 yang digunakan untuk menyimpan address ke dalam address latch.

-DTR/-R (data transmit/receive)

Diperlukan untuk mengatur arah pengiriman data bus controller.

-DEN (data enable)

Signal ini berfungsi agar data pada tranceiver dapat di-enable.

**-HOLD, HLDA**

Hold digunakan apabila sebuah prosesor lain ingin mempergunakan bus, dengan memerintahkan 8088 untuk melepaskan sistem bus, sehingga bus dapat dipakai oleh prosesor lain.

**-SSO (status line)**

Merupakan status line yang bersama-sama dengan IO/M dan DT/R membentuk suatu decoding dari bus cycle.

Tabel 2.6 Sistem decoding dari status bus cycle

IO/-M	DT/-R	-SSO	OPERASI
1	0	0	Interrupt acknowledge
1	0	1	Read I/O port
1	1	0	Write I/O port
1	1	1	Halt
0	0	0	Code access
0	0	1	Read memory
0	1	0	Write memory
0	1	1	Passive

#### II.4.2 PROGRAMABLE PERIPHERAL INTERFACE (8255)

PPI 8255 merupakan perangkat I/O yang luas penggunaannya, karena mudah pengoperasiannya dan kompatibel dengan

berbagai tipe mikroprosesor, yang berfungsi untuk menghubungkan komponen luar dengan sistem mikroprosesor. Dikatakan port I/O karena disinilah informasi keluar masuk dalam mikrokomputer, komponen I/O 8255 merupakan paralel I/O dan dapat diprogram untuk mentransfer data dalam berbagai kondisi dari I/O yang sederhana sampai dengan interrupt I/O yang komplek. 8255 dibentuk dalam 40 pin dual in line package dan dikelompokkan dalam tiga port yaitu port A, port B, port C dimana dipisah dalam 4 bit orde rendah dan 4 bit orde tinggi. Fungsi masing-masing port dapat diprogram sebagai input/output dengan cara menulis control word dalam control register.

Secara garis besar operasi PPI 8255 dapat diklasifikasikan dalam dua mode, yaitu mode set/reset (BSR) dan mode I/O. BSR digunakan untuk menset/reset bit dalam port, sedang mode I/O dibagi dalam 3 mode yaitu mode 0, mode 1 dan mode 2. Dalam mode 0 semua port berfungsi sebagai I/O sederhana, mode 1 adalah mode handshake dan mode 2 dimana periperal dapat digunakan sebagai port data bidirectional.

Secara blok diagram (lihat gambar), diperlihatkan PPI mempunyai dua kelompok 8 bit port A dan Port B dan kelompok 4 bit port C orde rendah dan orde tinggi, data bus buffer dan control logic. Pada bagian control logic terdapat 6 jalur pengontrol sebagai berikut:

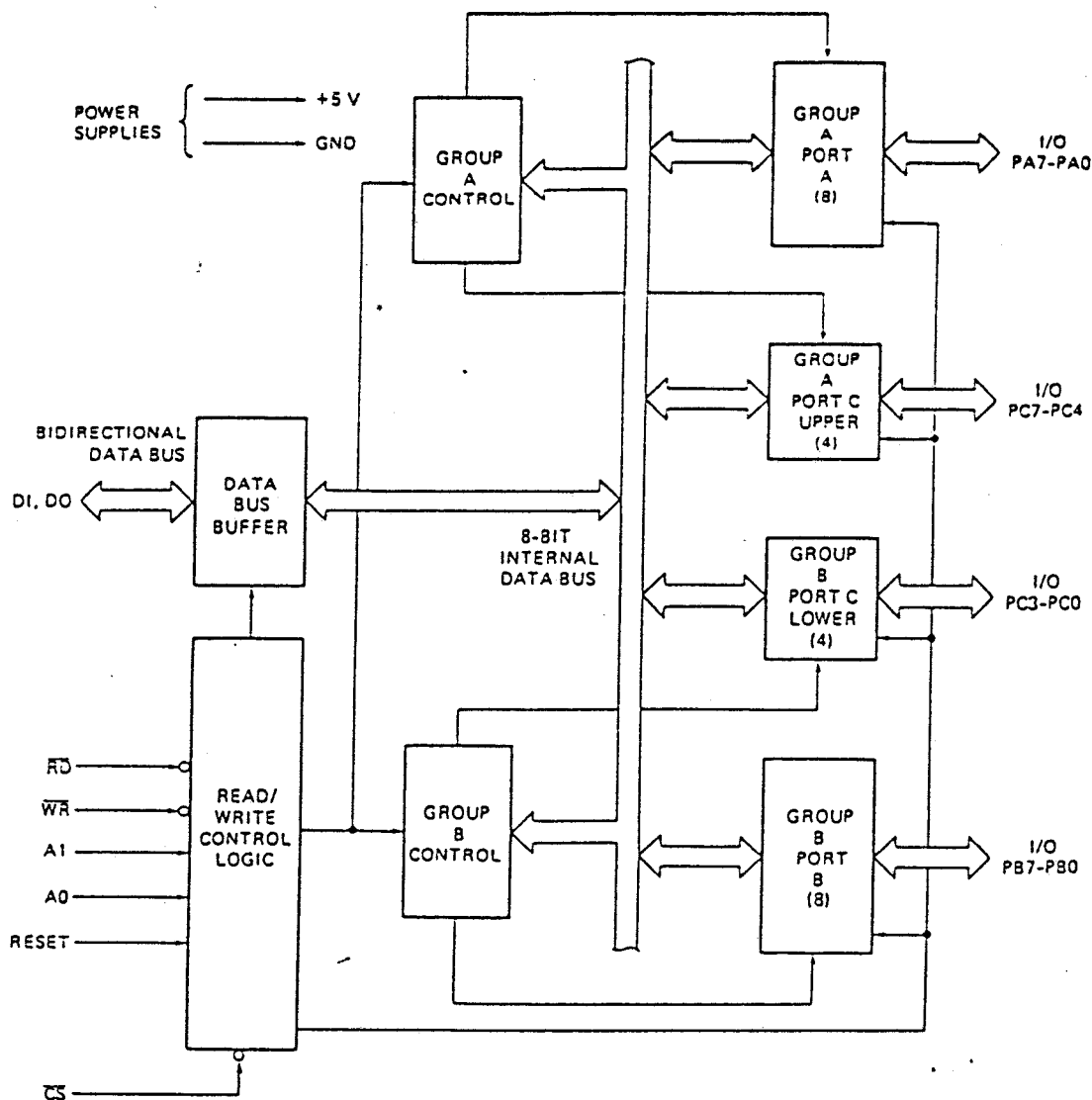
-RD (read)

Signal kontrol yang memungkinkan operasi baca, dimana jika

sinyal berlogika 0, maka mikroprosesor akan membaca port.

-WR (write)

Sinyal kontrol yang memungkinkan operasi penulisan dari I/O port yang dipilih.



Gambar 2.22 Blok diagram PPI 8255<sup>25</sup>

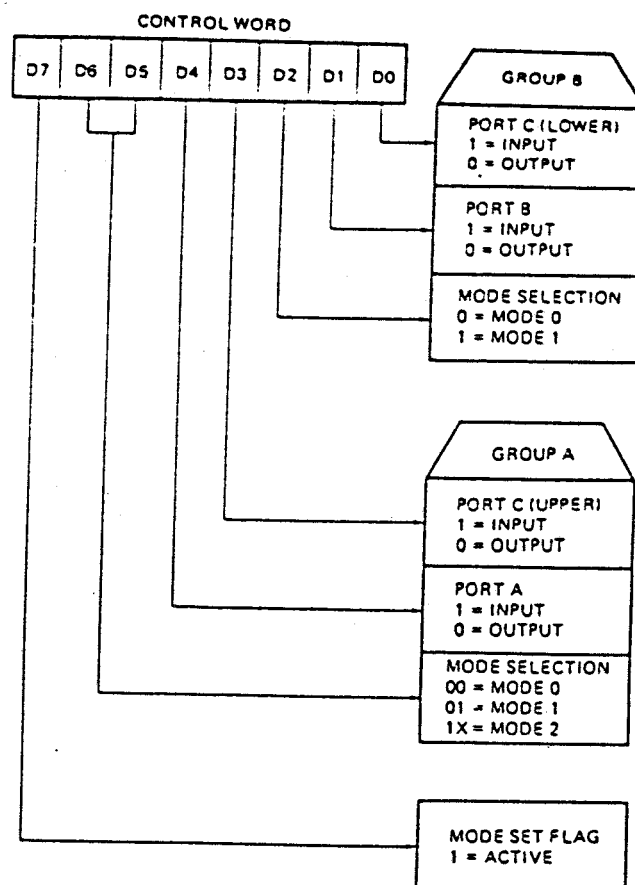
<sup>25</sup> Hall, Douglas V., Microprocessor And Interfacing Programming And Hardware, McGraw-Hill Book Company, Singapore, 1986, Hal. 263

### -RESET

Signal aktif tinggi, sinyal yang akan mereset control register dan menseting semua port ke dalam kode input.

### -CS (chip select)

Digunakan untuk mengaktifkan chip 8255, bila mendapat logika low mikroprosesor dapat mengakses data ke PPI 8255.



Gambar 2.23 Format control word 8255



#### -ADDRESS INPUT (A0-A2)

Kombinasi dari kedua address input input ini menentukan register/port dari 8255 yang akan menerima data dari atau ke mikroprosesor.

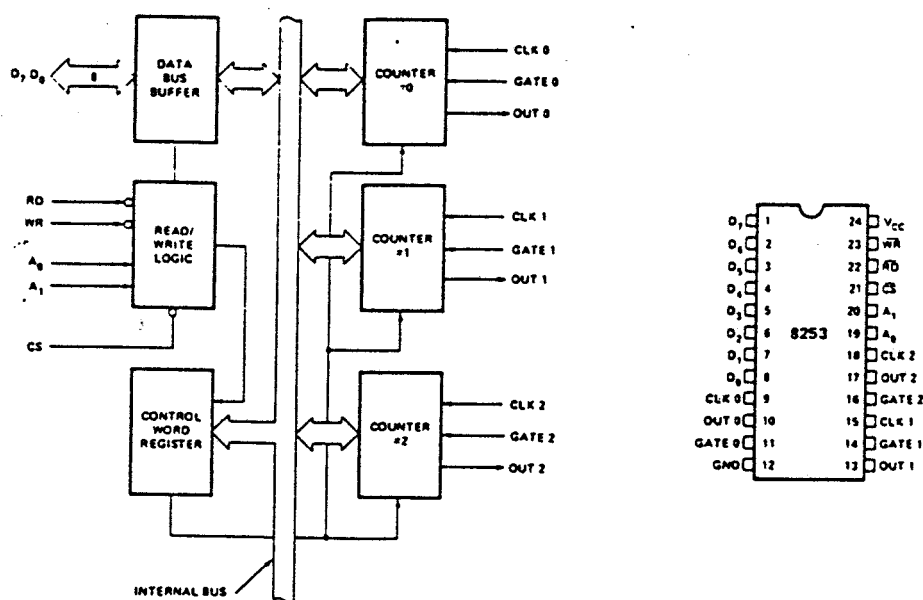
Tabel 2.7 Address pada PPI 8255

CS	A1	A0	YANG DIPILIH
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	CONTROL WORD
1	0	0	-

#### II.4.3. PROGRAMABLE INTERVAL TIMER 8253

8253 merupakan progamable interval timer ini dirancang untuk mengatasi persoalan kontrol waktu pada perancangan sistem mikrokomputer. Chip ini terdiri dari 16 bit counter yang terpisah satu sama lain dimana setiap counter dapat menerima clock sampai frekwensi 2,6 Mhz. Tiap counter dapat diprogram dan diproses dalam 6 mode. Diagram blok dari PIT 8253, terlihat pada gambar bahwa Pit mempunyai 3 counter yang dapat diprogram yang berdiri sendiri dan ketiga-tiganya identik, buffer bus sudah ada didalamnya sehingga dalam

penggunaanya tidak lagi diperlukan penyangga external



Gambar 2.24 Blok diagram PIT 8253<sup>26</sup>

Beberapa pin yang penting dari 8253, yaitu:

-DATA (D0-D7)

Bidirectional tri state bus data yang dihubungkan dengan bus data sistem mikroprosesor.

-ADDRESS INPUT (A0,A1)

Input ini digunakan untuk memilih salah satu dari counter atau kontrol word register untuk operasi write . atau read.

<sup>26</sup> Ibid., Hal. 21



-CS (chip select)

Logika low pada input ini akan mengaktifkan tanggapan 8253 terhadap sinyal RD dan WR. Jika diberikan logika tinggi maka 8253 disable terhadap sinyal RD maupun WR.

-RD (Read)

Control dari pembacaan data dari 8253. Input dari sinyal RD yang berlogika low pada operasi read.

-WR (write)

Input dari sinyal WR yang berlogika nol pada saat penulisan data oleh CPU ke 8253.

-CLK n

Clock input counter n.

-OUT n

Output dari counter n.

-GATE n

Input gate dari counter n.

Dalam PIT terdapat address internal yang akan menunjukkan pemilihan counter n atau control word, dibawah ini terdapat table yang berhubungan dengan hal tersebut.

Tabel 2.8 Address internal

A1	A0	SELECTS
0	0	COUNTER 0
0	1	COUNTER 1
1	0	COUNTER 2
1	1	CONTROL WORD REGISTER

Tabel 2.9 Format control

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

## SC - SELECT COUNTER:

SC1	SC0	
0	0	SELECT COUNTER 0
0	1	SELECT COUNTER 1
1	0	SELECT COUNTER 2
1	1	READ-BACK COMMAND (SEE READ OPERATIONS)

## RW - READ/WRITE:

RW1	RW0	
0	0	COUNTER LATCH COMMAND (SEE READ OPERATIONS)
0	1	READ/WRITE LEAST SIGNIFICANT BYTE ONLY.
1	0	READ/WRITE MOST SIGNIFICANT BYTE ONLY.
1	1	READ/WRITE LEAST SIGNIFICANT BYTE FIRST, THEN MOST SIGNIFICANT BYTE.

## M - MODE:

M2	M1	M0	
0	0	0	MODE 0 - INTERRUPT ON TERMINAL COUNT
0	0	1	MODE 1 - HARDWARE ONE-SHOT
X	1	0	MODE 2 - PULSE GENERATOR
X	1	1	MODE 3 - SQUARE WAVE GENERATOR
1	0	0	MODE 4 - SOFTWARE TRIGGERED STROBE
1	0	1	MODE 5 - HARDWARE TRIGGERED STROBE

## BCD:

0	BINARY COUNTER 16-BITS
1	BINARY CODED DECIMAL (BCD) COUNTER (4 DECADES)

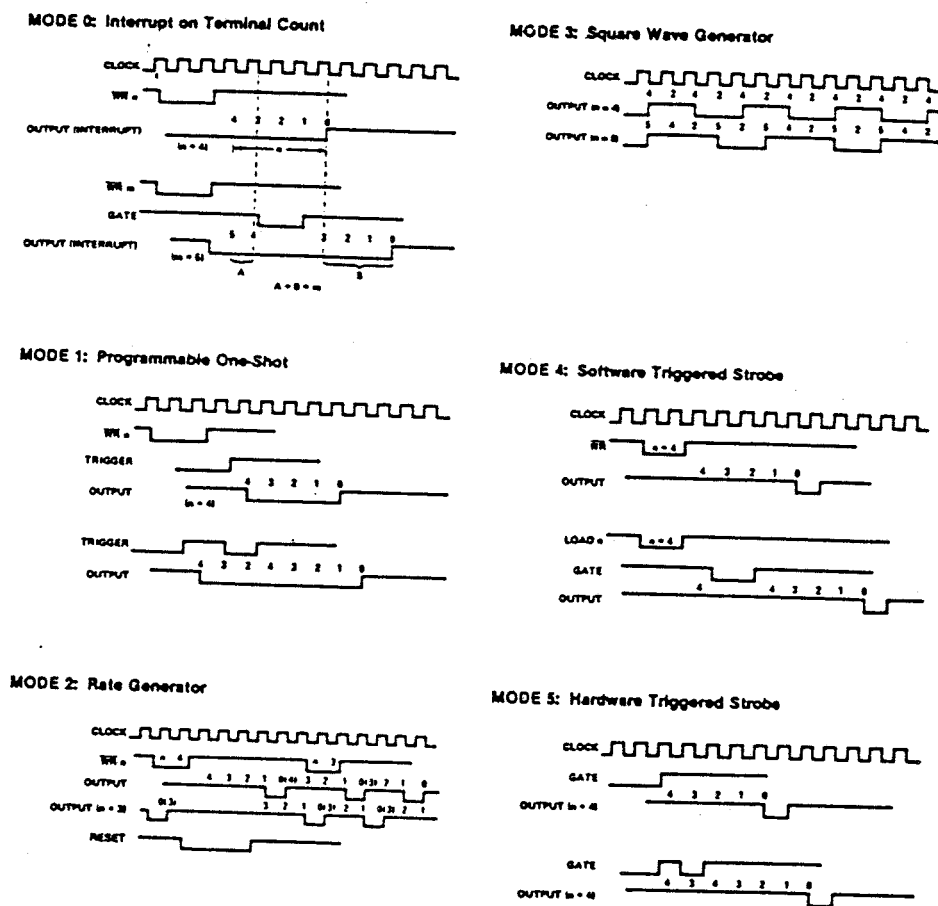
NOTE: DON'T CARE BITS (X) SHOULD BE 0 TO INSURE COMPATIBILITY WITH FUTURE INTEL PRODUCTS.

Progamable Interval Timer (8253) dapat dioperasikan dalam 6 mode. Adapun mode-mode itu adalah sebagai berikut:

1. Mode 0: Interupt pada terminal count.
2. Mode 1: Hardware-retriggerable single-shot.
3. Mode 2: Rate generator.
4. Mode 3: Square wave mode.
5. Mode 4: Software triggered strobe.

### 6. Mode 5: Hardware triggered strobe.

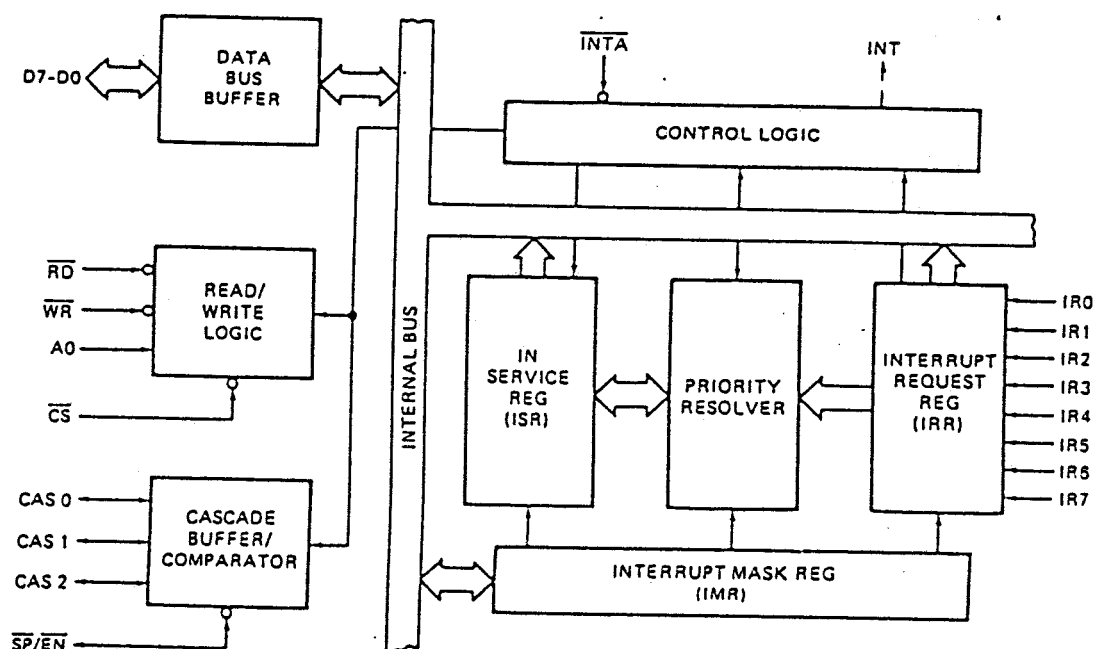
Gambar dibawah ini memperlihatkan PIT 8253 yang bekerja pada masing-masing mode. Dalam PIT juga terdapat fasilitas untuk membaca isi kounter dengan jalan melatch counter tanpa merubah isinya, kemudian besarnya hitungan dibaca.



Gambar 2.25 Diagram waktu PIT pada masing-masing mode

#### II.4.4. PROGRAMMABLE INTERRUPT CONTROLLER 8259

PIC 8259 merupakan chip yang digunakan untuk menghandel interrupt pada mikroprosesor, dengan IC ini maka kita dapat mengatur prioritas interrupt dari yang paling prioritas sampai yang terendah prioritasnya. Dalam diagram blok pada gambar terlihat dalam PIC terdapat bermacam-macam blok seperti ISR, IRR, IMR dan priority resolver. IRR berguna untuk menampung kondisi-kondisi input, IMR untuk mengenabel input register, ISR untuk menjaga input yang sedang dikerjakan, priority resolver untuk menentukan prioritas input.



Gambar 2.26 Blok diagram PIC 8259

Dalam PIC 8259 terdapat beberapa pin dan fungsi antara lain:

-VCC

Merupakan pin supply +5 V.

-GND

Merupakan ground.

-CS (chip select)

Logika 0 pada input ini akan menyebabkan PIC aktif. RD dan WR akan berkomunikasi dengan mikroprosesor.

-WR (write)

Logika 0 dan CS juga 0 akan mengakibatkan penulisan data dari mikroprosesor.

-RD (read)

Logika 0 dan CS juga 0 akan mengakibatkan pembacaan dari PIC ke mikroprosesor.

-D7-D0 (data)

Merupakan tempat bus data masuk dan keluar (bidirectional).

-CAS0-CAS2

Digunakan untuk pemasangan PIC secara bertingkat.

-INT (interrupt)

Pin ini akan 1 bila terjadi interrupt, pin ini dihubungkan pada pin INT pada mikroprosesor.

-IRO-IR7

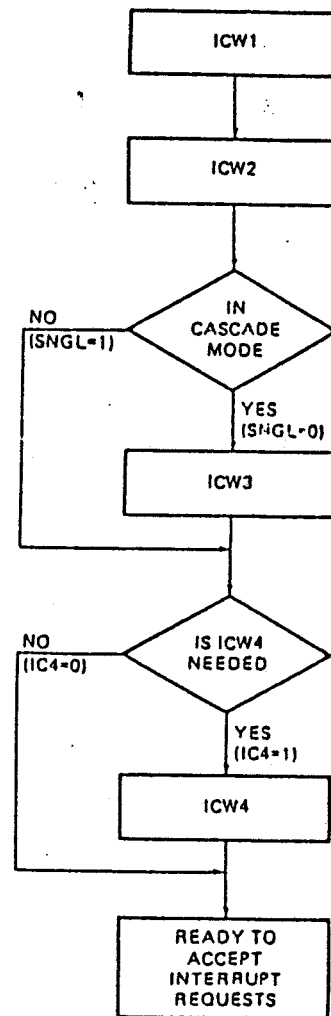
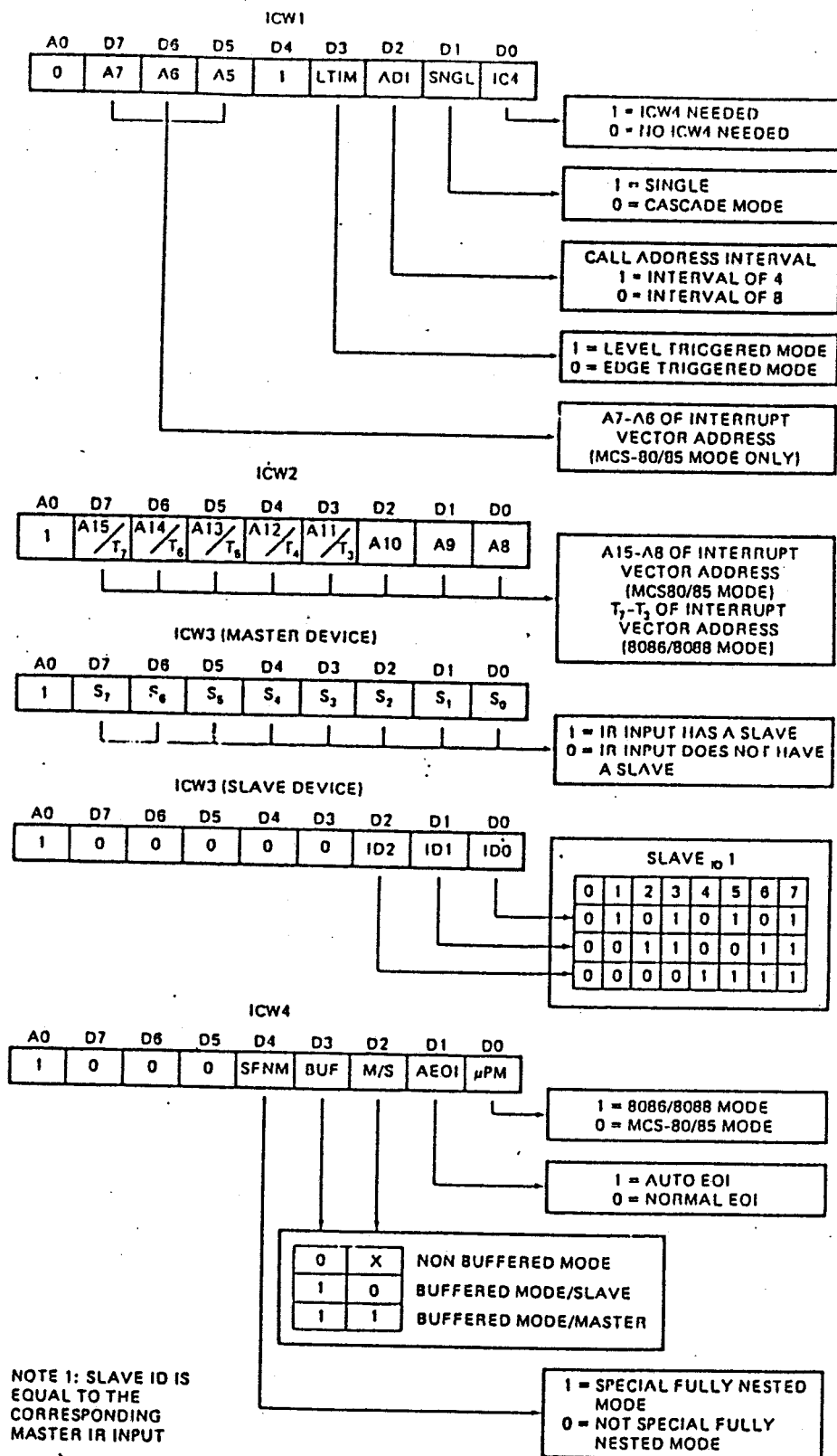
Merupakan pin dimana interrupt masuk

-INTA (interrupt acknowledge)

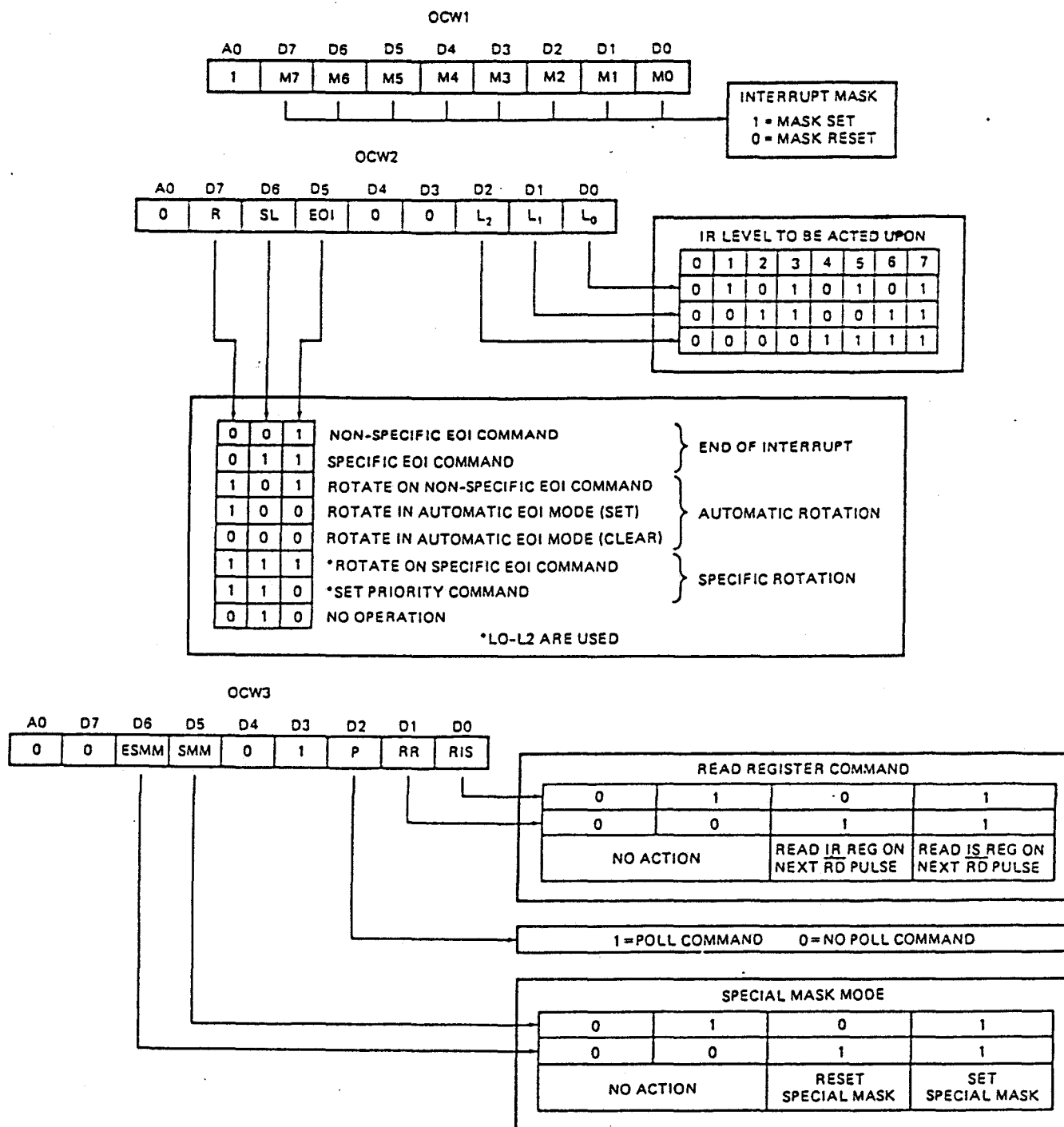
Pin ini digunakan PIC untuk mengenable interrupt vektor data.

-AO

Merupakan adress internal pada PIC.



Gambar 2.27 Format command word dan diagram alir inisialisasi



Gambar 2.28 Operasi dari Command Word 8259

Mode operasi pada PIC terdapat 6 jenis:

1. Fully nested loop.
2. Special fully nested loop.
3. Non specific rotating.
4. Special mask.
5. Specific rotating.
6. Pool.

Dalam pemrograman PIC ada beberapa cara yang harus diperhatikan antara lain:

1. Inisialisasi Command Word (ICW).
2. Operation command word

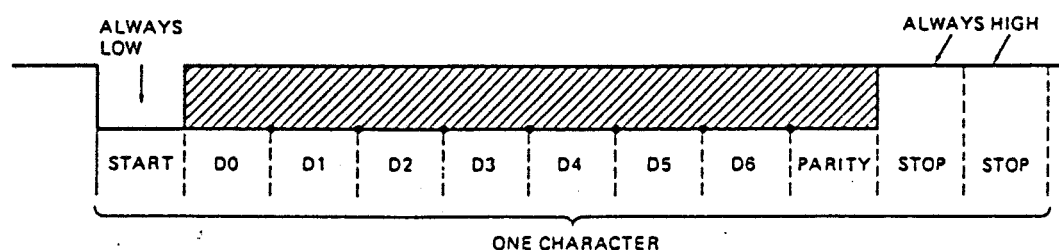
Untuk lebih jelasnya lihat gambar diagram alir cara inisialisasi PIC.

#### II.4.5. USART 8251

Pada sistem komputer data diolah secara paralel. Untuk transfer jarak jauh, data paralel kurang efisien karena membutuhkan jumlah kabel yang banyak dan level tegangan yang tidak memadai, oleh karena itu data perlu dirubah kedalam bentuk serial. Metoda transmisi data serial dapat dilakukan secara sinkron maupun asinkron. Pada transmisi sinkron data dikirim dalam blok dengan kecepatan tetap, awal dan akhir bluk diidentifikasikan dengan byte atau bit tertentu. Pada transmisi asinkron setiap karakter terdiri dari 1 yang mengidentifikasikan karakter dan satu 2 yang mengakhiri sebagai akhir karakter, karena setiap karakter diidentifi-



kasikansendiri-sendiri maka karakter dapat dikirim pada setiap saat (asinkron).

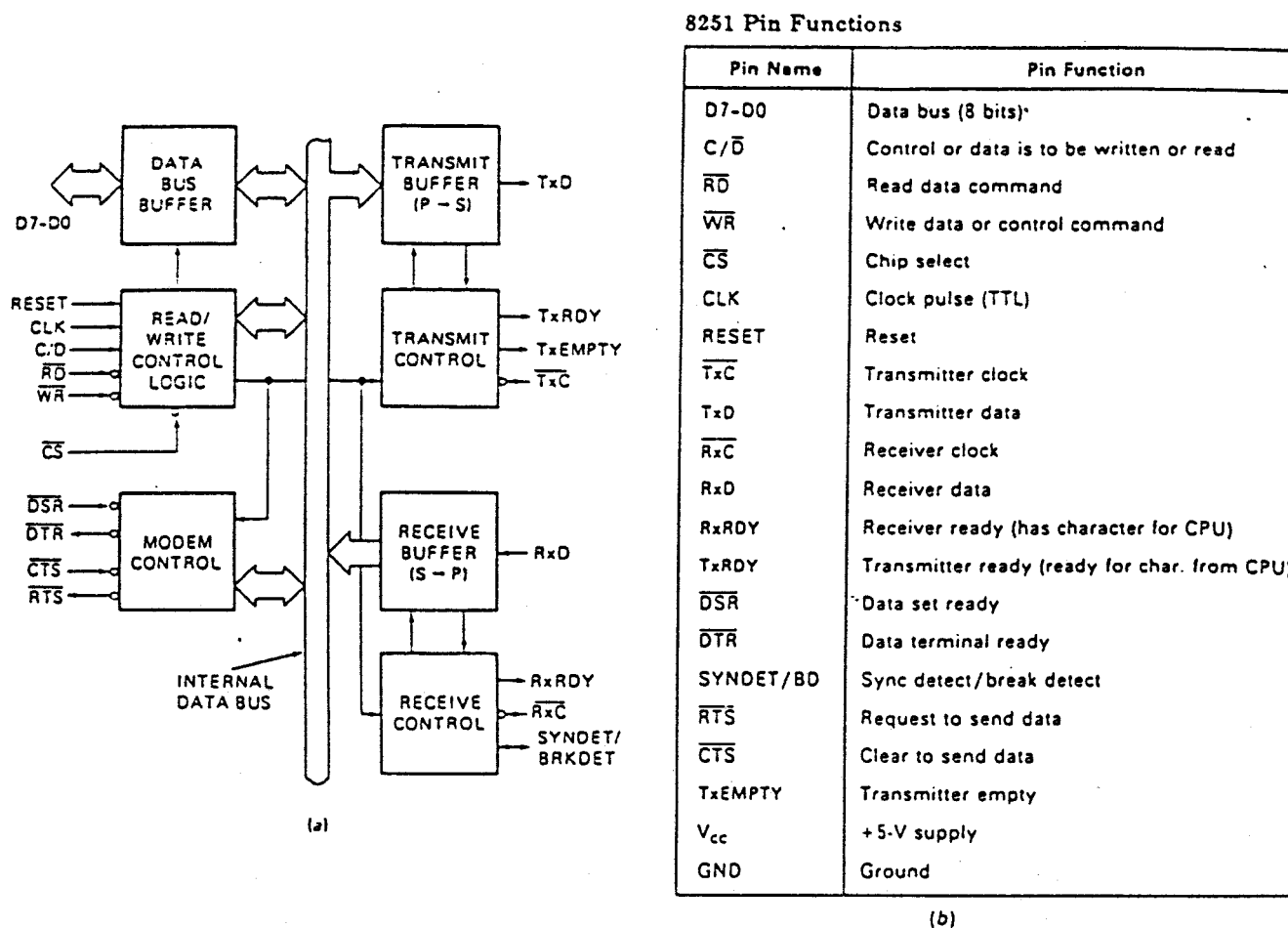


Gambar 2.29 Format bit transmisi serial asinkron

Dalam komunikasi serial mengenal, istilah baud rate yaitu kecepatan transfer data sebagai  $1/(\text{waktu 1bit})^{27}$

Untuk menghubungkan mikrokomputer dengan line data serial, data harus dikonversikan menjadi data serial, konversi dapat berupa paralel in-serial out dan serial in-paralel out register geser. Komponen yang digunakan untuk komunikasi serial data sinkron dan asinkron dinamakan USART. USART 8251 adalah komponen yang dibuat intel untuk mengimplementasikan konversi data paralel ke serial atau serial ke paralel, Usart 8251 kompatibel dengan keluarga mikroprosesor Intel seperti MCS-48, IAPX-86,88. CPU dapat membaca lengkap status USART pada setiap saat, termasuk kesalahan transmisi data dan sinyal kontrol seperti SYNDET dan TxEMPTY.

<sup>27</sup> Ibid., Hal. 443



Gambar 2.30 Blok diagram dan pin 8251<sup>28</sup>

Dari gambar terlihat berbagai bagian blok antara lain data-bus buffer yang berfungsi sebagai data buffer 8 bit data bus dari mikrokomputer, dan bersifat bidirectional data, read/write control logic berfungsi sebagai pengendali perpindahan dan common status, sehingga memungkinkan data dapat dibaca maupun ditulis pada lokasi yang tepat, modem kontrol berfungsi sebagai perantara hubungan 8251 dengan modem,

<sup>28</sup>Ibid., hal 490

receiver buffer dan receiver control dua blok ini berfungsi secara bersama-sama dalam mentransmisikan data serial.

Pada 8251 terdapat 28 pin dengan fungsi masing-masing sebagai berikut:

**-D0-D7**

Merupakan jalur data 8 bit yang digunakan untuk hubungan dengan jalur data sistem mikroprosesor, semua informasi termasuk inisialisasi pemrograman word register dilewatkan melalui jalur ini.

**-RESET**

Logika high pada pin ini menyebabkan 8251 dalam keadaan idle.

**-CLOCK**

Untuk mensinkronisasi semua operasi dalam USART.

**-WR (write)**

Input low pada pin ini menyebabkan mikroprosesor menuliskan data atau kontrol word ke USART.

**-RD (read)**

Input low pada pin ini menyebabkan mikroprosesor menulis data atau status dari USART 8251.

**-C/-D**

Masukan pada pin ini berfungsi memberi informasi pada USART 8251A, jenis data yang ada pada data bus, data atau control word.

**-CS (chip select)**

USART 8251 diaktifkan dengan memberikan logika low pada pin ini. Logika high menyebabkan bus data dalam keadaan mengambang.

-DSR (data set ready)

Pin masukan ini dapat dideteksi oleh mikrokomputer. Biasanya digunakan untuk menguji keadaan modem.

-DTR (data terminal ready)

Pin keluaran ini biasanya digunakan untuk data terminal ready.

-CTS (clear to send)

Logika low pada pin ini menyebabkan 8251A dapat mengirim data serial, jika bit TXEn pada command word diaktifkan, pin ini diperlukan untuk handshaking.

-RTS (request to send)

Pin ini merupakan jalur keluaran bagi data serial yang akan dikirimkan.

-TXD (transmit data)

Pin ini merupakan jalur keluaran bagi data serial yang akan dikirim

-RXD (receive data)

Pin ini merupakan jalur masukan, dimana data serial diterima.

-TXC (transmitter clock)

Pin masukan bagi clock yang diperlukan untuk mengendalikan baud rate pengiriman.

-RXC (receiver clock)

Pin masukan bagi clock yang diperlukan untuk mengendalikan baud rate penerimaan data.

-TXRDY

Untuk memberitahu bahwa pada penerima telah siap dikirim mikroprosesor.

-RXRDY (receiver ready)

Untuk memberitahu bahwa pada penerima telah siap dibaca mikrokomputer.

-Sydet/BD (syn detect/break detect)

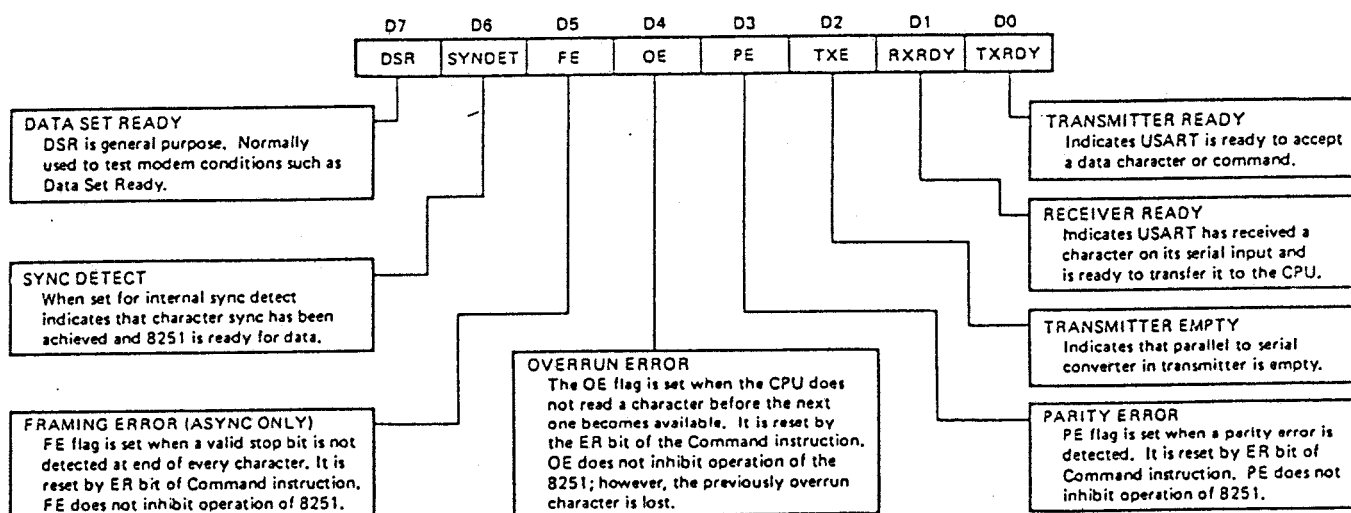
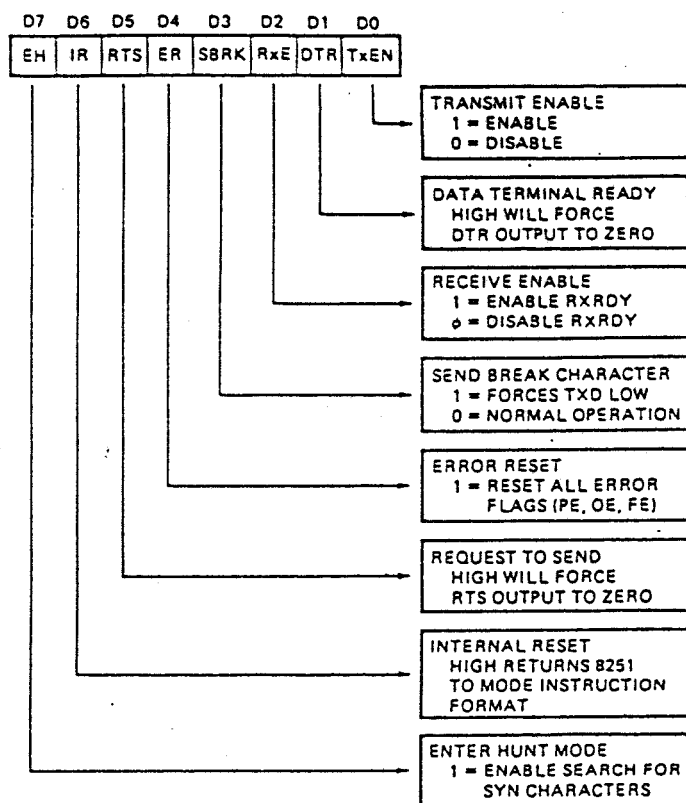
Mempunyai dua fungsi:

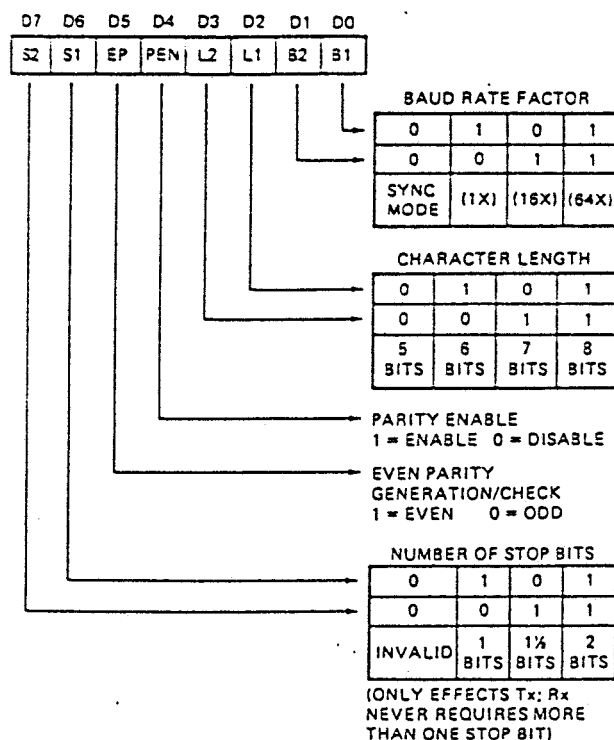
1. Pada mode sinkron, bila input data RXD berada dalam keadaan 0 selama lebih dari dua waktu karakter, pin ini akan high.
2. Pada mode asinkron pin ini akan high bila 8251A mendeteksi adanya karakter sinkronisasi pada deret bit data.

Untuk menjalankan sistem ini, 8251 harus diprogram dulu dengan jalan menginisialisasi. Inisialisasi dilakukan oleh mikroprosesor dengan jalan mengirim kontrol word yaitu:

1. Mode instruction.
2. Command instruction.

Sedangkan untuk mengetahui status dari 8251A digunakan status word dari 8251A. (lihat gambar).

Gambar 2.31 Bentuk mode word dan status word 8251A<sup>29</sup><sup>29</sup>Ibid., Hal. 491



Gambar 2.32 Bentuk command word 8251A

Pada komunikasi asinkron ini menggunakan standar RS 232C dimana mempunyai karakteristik sebagai berikut:

1. Tegangan rangkaian terbuka maximum 25 V.

-Keadaan logika 1 ditandai dengan tegangan yang lebih negatif dari -3 V. Pada logika 1 besarnya tegangan harus antara -5V dan -15V.

-Keadaan logika 0 ditandai dengan tegangan yang lebih positif dari +3V. Pada logika 0 besarnya tegangan antara +5V dan +15V.

2. Slew rate (perubahan tegangan keluaran persatuan waktu) harus lebih kecil dari 30V/mikro-detik. Waktu peralihan melewati daerah peralihan -3 V sampai +3 V tidak lebih dari 1 milidetik.
3. Digunakan konektor penghubung DB-25 atau DB-9.

PIN NUMBERS FOR 9 PINS	PIN NUMBERS FOR 25 PINS	COMMON NAME	RS-232C NAME	DESCRIPTION	SIGNAL DIRECTION ON DCE
3	1		AA	PROTECTIVE GROUND	-
2	2	TXD	BA	TRANSMITTED DATA	IN
7	3	RXD	BB	RECEIVED DATA	OUT
8	4	RTS	CA	REQUEST TO SEND	IN
	5	CTS	CB	CLEAR TO SEND	OUT
6	6	DSR	CC	DATA SET READY	OUT
5	7	GND	AB	SIGNAL GROUND (COMMON RETURN)	-
1	8	CD	CF	RECEIVED LINE SIGNAL DETECTOR	OUT
	9		-	(RESERVED FOR DATA SET TESTING)	-
	10		-	(RESERVED FOR DATA SET TESTING)	-
	11		SCF	UNASSIGNED	-
	12		SCB	SECONDARY RECEIVED LINE SIGNAL DETECTOR	OUT
	13		SBA	SECONDARY CLEAR TO SEND	OUT
	14		DB	SECONDARY TRANSMITTED DATA	IN
	15			TRANSMISSION SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
	16		SBB	SECONDARY RECEIVED DATA	OUT
	17		DD	RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE)	OUT
	18			UNASSIGNED	-
4	19		SCA	SECONDARY REQUEST TO SEND	IN
	20	DTR	CD	DATA TERMINAL READY	IN
9	21		CG	SIGNAL QUALITY DETECTOR	OUT
	22		CE	RING INDICATOR	OUT
	23		CH/CI	DATA SIGNAL RATE SELECTOR (DTE/DCE SOURCE)	IN/OUT
	24		DA	TRANSMIT SIGNAL ELEMENT TIMING (DTE SOURCE)	IN
	25			UNASSIGNED	-

Gambar 2.33 Conector DB-25 dan DB-9



### BAB III

#### PERENCANAAN ALAT

Pada bab ini akan dijelaskan tentang perencanaan alat dari sistem yang dibuat, untuk perencanaan pada sistem ini secara garis besar dibagi menjadi 2 bagian:

1. Perencanaan perangkat keras.
2. Perencanaan perangkat lunak.

Pada perencanaan perangkat keras akan meliputi penjelasan dari perancangan diagram blok sistem, perancangan sensor-sensor beserta pengkondisi sinyal, perancangan minimum sistem 8088, perancangan peripheral yang digunakan (PPI, PIT, PIC, USART, ADC).

Pada perancangan perangkat lunak akan meliputi penjelasan dari perangkat lunak yang digunakan pada sistem minimum 8088 dan perangkat lunak yang digunakan pada perangkat IBM PC.

#### III.1. PERENCANAAN PERANGKAT KERAS

Pada perencanaan perangkat keras akan dibagi beberapa bagian sebagai berikut:

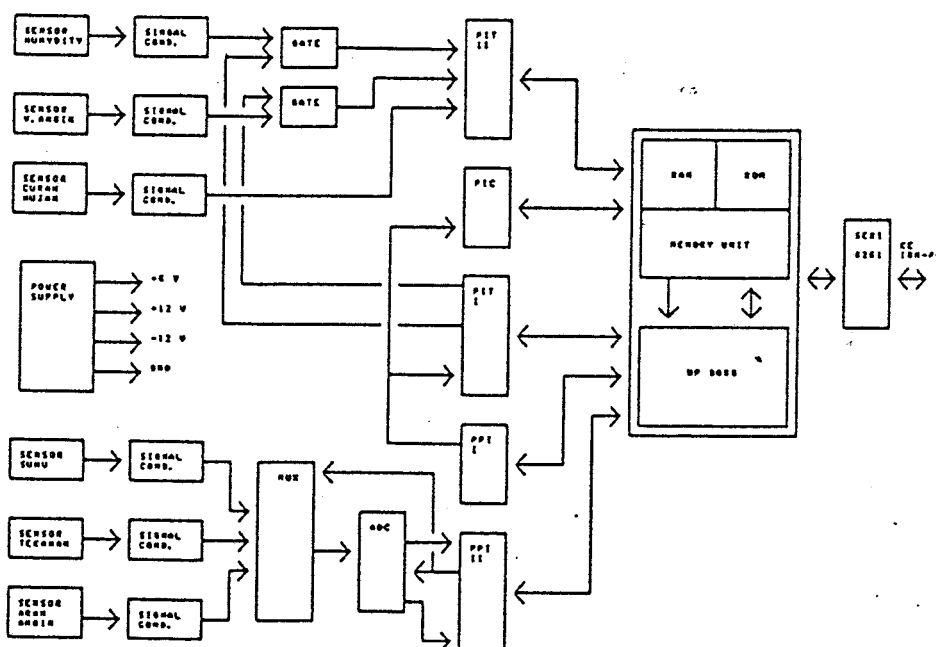
##### III.1.1. PERENCANAAN DIAGRAM BLOK

Diagram blok sistem meliputi beberapa bagian pokok diantaranya adalah:

1. Rangkaian uP 8088 pada mode minimum sebagai unit

pengendali dan pemroses data, dengan dibantu beberapa komponen lain.

2. Rangkaian periperal sebagai pembantu untuk pengambilan data dari sensor, seperti PPI, PIT, PIC, USART.
3. Rangkaian ADC, beserta komponen penunjangnya seperti multiplexer, tegangan referensi.
4. Rangkaian sensor beserta beserta pengkondisi sinyal.  
Dalam hal ini terdapat 6 sensor yang dipakai yaitu: sensor kelembaban, sensor kecepatan angin, sensor curah hujan, sensor suhu, sensor tekanan, sensor arah angin.
5. Rangkaian power supply, sebagai pemberi daya.



Gambar 3.1 Diagram blok sistem

### III.1.1.1. PRINSIP KERJA ALAT

Dalam sistem ini terdapat dua besaran yang diukur yaitu:

1. Besaran frekwensi, meliputi kelembaban, kecepatan angin, curah hujan.
2. Besaran tegangan, meliputi suhu, tekanan, arah angin.

Dasar kerja dari masing-masing sensor sampai mendapatkan informasi data yang diinginkan adalah sebagai berikut:

-Besaran kelembaban diperoleh dari sensor kelembaban yang berupa kapasitor dimana kapasitansinya besarnya akan berubah sehubungan dengan berubahnya kelembaban udara, efek tersebut dipadukan dengan suatu astable multivibrator, yang akan mempunyai keluaran frekwensi tertentu, sesuai dengan kelembaban yang ada, dan besar frekwensi keluaran tersebut akan diukur sistem dengan cara melewati pada sebuah gate yang akan membuka dengan lama waktu 1 detik sehingga besarnya frekwensi yang terukur langsung dalam satuan Hz. Untuk menghitung banyaknya frekwensi yang masuk dan pengaturan gerbang untuk dibuka selama 1 detik dilakukan oleh PIT (sebagai monostable dan penghitung), serta PPI (sebagai pemberi trigger).

-Besaran kecepatan angin diperoleh dari sensor mekanik yang berbentuk mangkok, dimana bila tertiup angin sensor tersebut akan berputar dan menghasilkan pulsa tertentu sesuai dengan kecepatan angin yang bertiup. Besarnya frekwensi

yang ada diukur dengan melewati pada sebuah gate yang dapat membuka selama 1 detik, sehingga frekwensi yang terukur berskala Hz, Untuk menghitung besarnya frekwensi yang masuk dan pengaturan membuka gate selama 1 detik digunakan dengan PIT (sebagai monostable dan penghitung), serta PPI (sebagai pemberi trigger).

-Besaran curah hujan diperoleh dengan menggunakan sensor mekanik curah hujan, yang pada hakekatnya merupakan sebuah alat penimbang yang menghasilkan pulsa untuk dihitung, banyaknya pulsa yang masuk berhubungan dengan banyaknya air hujan yang tertampung. Untuk menghitung banyaknya pulsa yang terjadi digunakan PIT sebagai penghitung.

-Besaran suhu diperoleh dengan menggunakan sensor suhu (IC LM 335), dimana mempunyai koefisien tertentu ( $V/^{\circ}K$ ) yang keluarannya sebelum dikonversi oleh ADC dilewatkan pada pengkondisi sinyal (untuk dikuatkan), dan dimultipleks agar dapat dibaca oleh ADC sesuai dengan kanal masukan. Koordinasi dari kanal, sinyal kontrol ADC, pembacaan data ADC dilakukan oleh PPI.

-Besaran tekanan diperoleh dengan menggunakan sensor tekanan yang berupa resistor jembatan (SCC 15 A) yang mempunyai koefisien "perubahan  $R/Inc-Hg$ " tertentu. Untuk menghasilkan tegangan keluaran yang dapat dibaca oleh ADC, maka perlu diberi pengkondisi sinyal berupa sumber arus dan penguat. Besarnya tegangan yang akan dikonversi oleh ADC sebelumnya dimultipleks agar bersesuaian dengan kanal yang ada.

-Besaran arah angin diperoleh dengan menggunakan sensor mekanik yang berbentuk sirip, dimana putaran sirip tersebut akan menunjukkan arah angin bertiup, dalam sirip tersebut putaran yang terjadi karena tiupan angin diubah menjadi parameter tegangan oleh suatu potensiometer khusus (dengan diberi sumber daya tetap) yang dapat berputar  $360^\circ$ , sehingga tegangan yang ada akan dikonversi oleh ADC sesuai besar dari derajat putarannya. Sebelum keluaran tegangan dikonversi oleh ADC diperlukan suatu multiplexer agar bersesuaian dengan kanal yang ada.

Setelah keenam parameter tersebut didapat, maka data yang siap untuk dikirim melalui fasilitas komunikasi seri (USART) RS-232C ke IBM-PC. Dari semua yang dijelaskan diatas, masing-masing sub bagian blok dikoordinasi oleh perangkat minimum sistem 8088.

### III.1.2. PERENCANAAN SENSOR PARAMETER DAN PENGKONDISI SINYAL

Pada bagian ini terdapat 6 sensor beserta pengkondisi sinyalnya, yaitu

1. Bagian sensor kelembaban dan pengkondisi sinyal.
2. Bagian sensor kecepatan angin dan pengkondisi sinyal.
3. Bagian sensor curah hujan dan pengkondisi sinyal.
4. Bagian sensor suhu dan pengkondisi sinyal.
5. Bagian sensor tekanan dan pengkondisi sinyal.
6. Bagian sensor arah angin dan pengkondisi sinyal.

### III.1.2.1. PERENCANAAN SENSOR KELEMBABAN DAN PENGKONDISI SINYAL

Pada bagian ini direncanakan untuk mengukur besarnya kelembaban yang ada pada udara sekitarnya. Sensor yang dipakai adalah sensor yang berbentuk kapasitor dimana besar kapasitansinya akan berubah-ubah terhadap kelembaban udara sekitarnya, besarnya kapasitansi dan kelembaban udara dapat dilihat dalam lampiran data. Secara typical besarnya kapasitansi berharga sebesar 112 pF pada saat kelembaban 7,5 % dan bergerak secara parabolik hingga kapasitansinya sebesar 143,5 pF pada kelembaban 90%. Perubahan kapasitansi ini dijadikan dasar prinsip untuk mengukur kelembaban. Pada perancangan ini, sensor kelembaban akan dipadukan dengan suatu astable multivibrator LM 555(sebagai pengkomdisi sinyal), sehingga akan berfungsi sebagai perubah besaran kapasitansi menjadi besaran frekwensi, dimana frekwensi keluarannya akan dihitung oleh sebuah counter.

Sesuai dengan data pada lampiran sensor kelembaban ini bekerja pada frekwensi 1Khz-1Mhz, maka dengan menggunakan IC LM 555 akan diperoleh perhitungan sebagai berikut:

$$F = \frac{1}{T}$$

$$T = 0,693 (R1 + 2R2) C, \text{ ambil } R1 = R2 = 100K, C \text{ awal} = 112 \text{ pF}$$

$$T1 = 0,693 (100K + 200K) 112 \text{ pF} = 23,18 \text{ uS}$$

$$F1 = 42946,47 \text{ Hz (batas awal)}$$

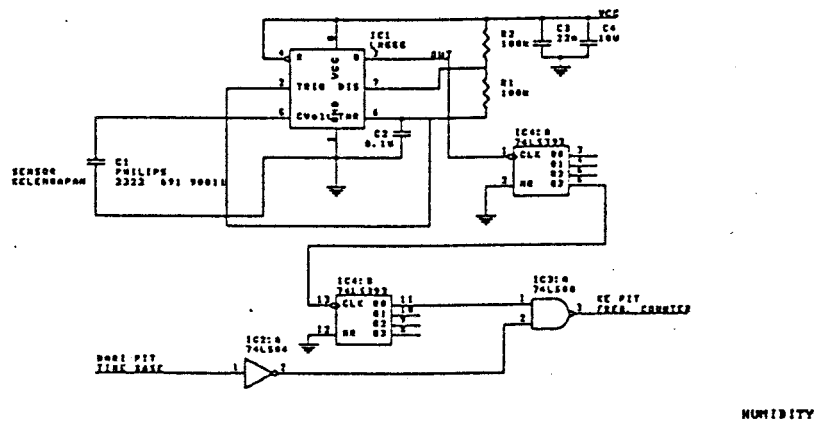
$$T2 = 0,693 (R1 + 2R2) C, C \text{ akhir} = 143,5 \text{ pF}$$



$$T2 = 0,693 (100K + 200K) 143,5pF = 29,83 \mu s$$

$$F2 = 33519,19 \text{ Hz (batas akhir)}$$

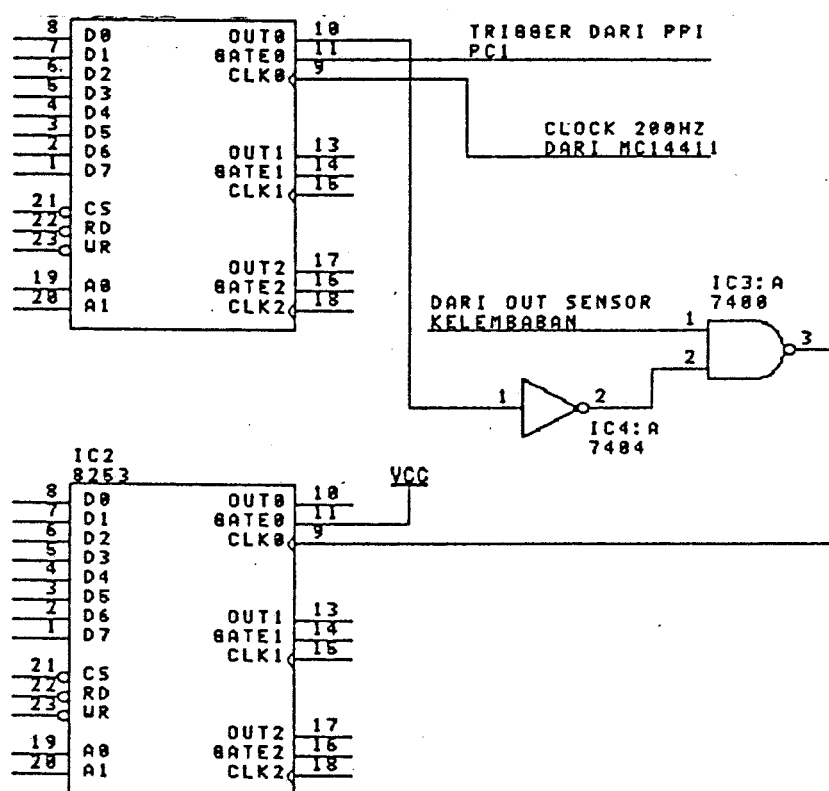
Jadi dengan harga diatas memenuhi untuk daerah frekwensi kerja sensor, sebelum keluaran dari astable diukur oleh counter frekwensi keluaran dibagi sebesar 32, dengan menggunakan 74LS393 sehingga tidak terjadi range yang terlalu besar. Lihat gambar 3.2.



Gambar 3.2 Rangkaian sensor kelembaban dan astable multivibrator.

Untuk bagian counter dirancang dengan menggunakan prinsip frekwensi counter yaitu memanfaatkan suatu gerbang yang dapat membuka dan menutup pada durasi waktu selama 1 detik. Untuk koordinasi membuka dan menutup selama 1 detik digunakan suatu Programable Interval Timer (PIT) yang

bekerja pada mode 1 (monostable), sehingga besarnya frekwensi yang dihitung beskala Hz. Untuk interval selama 1 detik PIT diinisialisasi sehingga besarnya  $N * \text{Clock}$  akan sebesar 1 detik, sedangkan untuk menghitung besarnya frekwensi yang dikeluarkan oleh gerbang digunakan PIT yang bekerja pada mode 0, dimana counter diinisialisasi sebesar FFFFH.



Gambar 3.3 Penghitung frekwensi dengan PIT 8253

### III.1.2.2. PERENCANAAN BAGIAN SESNSOR KECEPATAN ANGIN DAN PENGKONDISI SINYAL

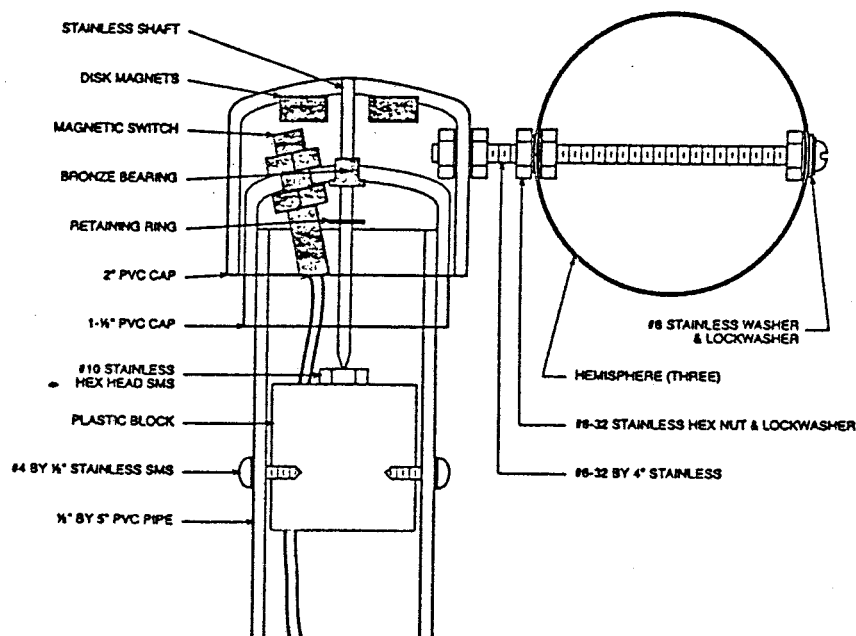
Pada perencanaan ini dibagi menjadi 2 bagian yaitu:



1. Perencanaan bagian mekanik.
2. Perencanaan bagian elektronik.

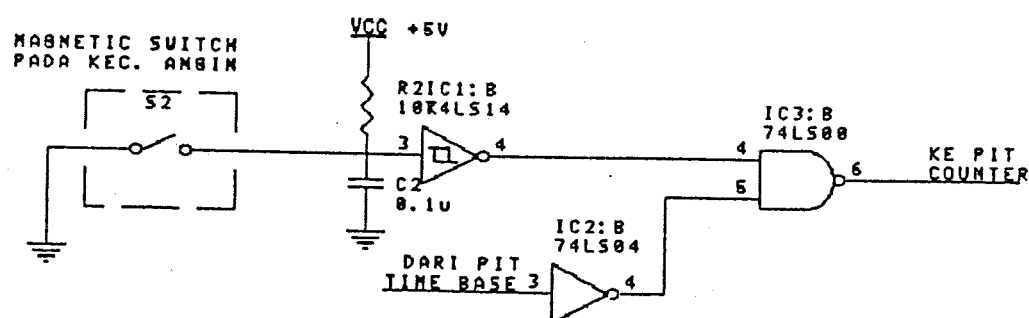
Pada perancangan bagian mekanik dibuat suatu bentuk anemometer mangkok, yang akan merubah kecepatan translasi menjadi kecepatan angular, dalam alat ini digunakan mangkok yang berjumlah tiga buah dengan selang sudut sebesar  $120^\circ$ , untuk putaran yang seimbang digunakan sebuah pipa PVC serta bantalan untuk berputar pada satu poros, sedangkan pada bagian atas (penutup pipa PVC) merupakan tempat meletakkan lempengan magnetik yang berguna mengatur buka dan tutupnya magnetik switch. untuk lebih jelasnya lihat gambar 3.4.

## ANEMOMETER



Gambar 3.4 Bagian mekanik sesnsor kecepatan angin

Pada perancangan bagian elektronik yang berfungsi sebagai pengkondisi sinyal adalah suatu magnetik switch yang bekerja sebagai pembangkit pulsa. Magnetik switch mempunyai kedudukan normal pada posisi off sedangkan akan on bila ada medan magnet yang memotongnya, dari prinsip dasar tersebut maka putaran dari sensor akan digunakan untuk memotong dari magnetik switch, sehingga akan terjadi kejadian on dan off dari magnetik switch, yang akan dihitung frekwensinya. Untuk menghilangkan efek bouncing, dikombinasi dengan rangkaian R dan C serta IC schmitt trigger. Sedangkan pulsa yang dihasilkan akan dihitung dengan counter untuk menentukan berapa besar frekwensi yang dihasilkan. Lihat gambar 3.5.



Gambar 3.5 Rangkaian sensor kecepatan angin

Untuk bagian counter dirancang dengan menggunakan prinsip frekwensi counter yaitu memanfaatkan suatu gerbang yang dapat membuka dan menutup pada durasi waktu selama 1 detik. Untuk koordinasi membuka dan menutup selama 1 detik digunakan suatu Programable Interval Timer (PIT) yang bekerja pada mode 1 (monostable), sehingga besarnya frekwensi yang dihitung beskala Hz. Untuk interval selama 1 detik PIT diinisialisasi sehingga besarnya  $N * \text{Clock}$  akan sebesar 1 detik, sedangkan untuk menghitung besarnya frekwensi yang dikeluarkan oleh gerbang digunakan PIT yang bekerja pada mode 0, dimana counter diinisialisasi sebesar FFFFh dan setelah 1 detik besarnya hitungan mundur dari counter tersebut diambil sebagai besarnya frekwensi yang terukur. Lihat gambar 3.3 (sesuai dengan prinsip menghitung besarnya frekwensi pada sensor kelembaban).

#### III.1.2.3. PERENCANAAN SENSOR CURAH HUJAN DAN PENGKONDISI SINYAL.

Pada bagian sensor curah hujan ini mempunyai prinsip utama sebagai penghitung banyaknya air yang tertampung dalam sensor tersebut. Perencanaan bagian ini dibagi menjadi 2 yaitu:

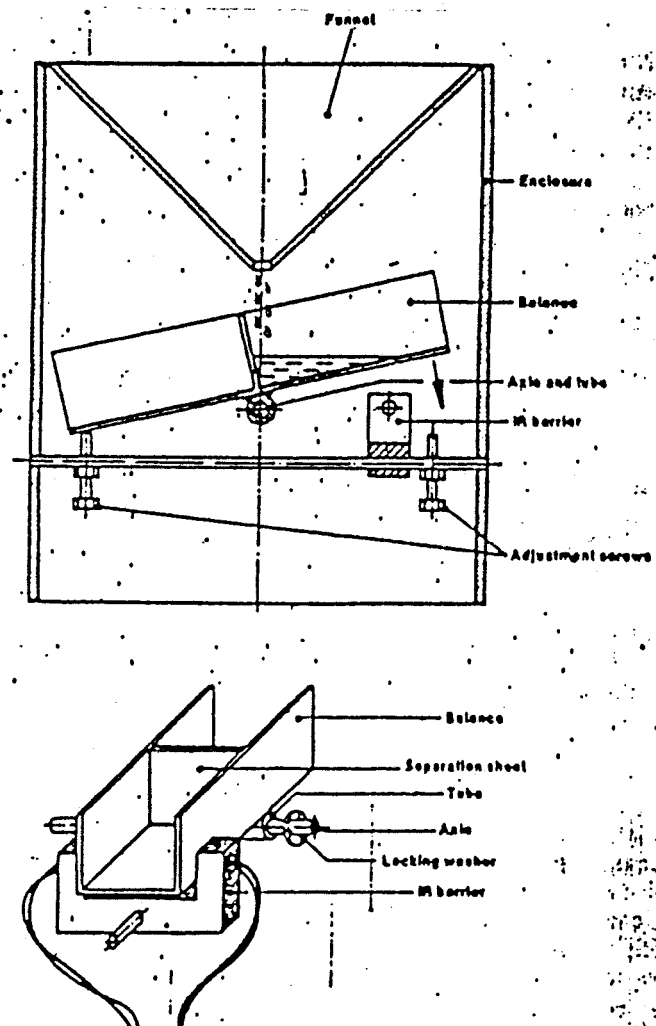
1. Perencanaan bagian mekanik.
2. Perencanaan bagian elektronik.

Pada bagian mekanik dirancang sesuai prinsip kerja utama sensor ini terletak dalam alat penakar air yang berbentuk

seperti timbangan, dimana mempunyai bentuk profil seperti huruf U dan terbuat dari metal, alat tersebut akan menumpahkan cairan yang tertampung pada saat volume tertentu, karena adanya gerakan naik turun timbangan maka volume air yang tertampung akan terhitung dengan menentukan berapa kali timbangan itu bergerak naik turun dan berapa volume air yang ditumpahkan oleh timbangan pada sekali gerakan. Banyaknya air yang ditumpahkan untuk sekali gerakan merupakan resolusi terkecil dari sensor curah hujan ini. Untuk lebih jelasnya lihat gambar 3.6. Pada sensor terlihat beberapa bagian antara lain adalah tabung pembungkus, corong bagian atas dengan diameter 18 Cm, timbangan mekanik dengan pemberat tertentu, skrup untuk menyetel keseimbangan dari timbangan, saklar magnetik, poros dari timbangan beserta bantalannya.

Untuk besarnya resolusi yang ada, dapat ditentukan sebagai berikut:

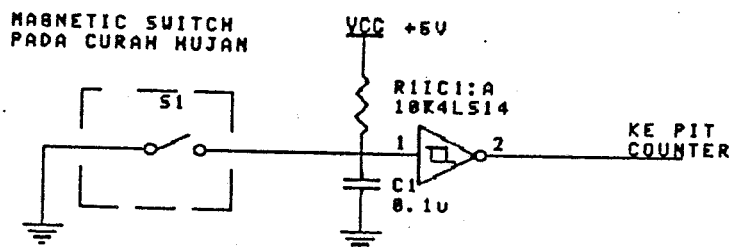
- Dari definisi 10 mm curah hujan akan sepadan dengan volume air A ml, dimana A adalah besarnya luas dari lingkaran corong yang digunakan dalam  $\text{Cm}^2$
- Dirancang suatu sensor curah hujan dengan resolusi 1 mm.
- Besar diameter dari corong  $D = 18 \text{ Cm}$ , maka luas dari corong  $L = (0,5 D)^2 \pi = (0,5 \cdot 18)^2 \cdot 3,14 = 254 \text{ Cm}^2$ .
- Jadi untuk volume 254 ml harus terjadi 10 kali gerakan naik turun (menghitung sampai 10mm ).



Gambar 3.6 Bagian mekanik sensor curah hujan

Pada perancangan bagian elektronik, ditujukan untuk menghitung banyaknya tetesan air hujan yang tertampung, maka kita harus menghitung banyaknya gerakan naik turun dari sensor yang digunakan, salah satu cara yaitu dengan menggunakan saklar magnetic, dimana alat tersebut mempunyai kea-

daan normal terbuka dan akan tertutup bila ada medan magnet yang memotongnya. Untuk membangkitkan medan magnet digunakan suatu lempengan magnet yang diletakkan pada salah satu dinding ujung timbangan, sehingga bila ada gerakan timbangan naik turun maka akan menyebabkan saklar tersebut on dan off dan akan menghasilkan pulsa yang jumlahnya menyatakan banyaknya gerakan naik turun, untuk menghindari adanya bouncing maka saklar tersebut dikombinasi dengan anti bouncing (R dan C) serta schmitt trigger 74LS14. Lihat gambar 3.7.



Gambar 3.7 Rangkaian sensor curah hujan

Dari pulsa yang dihasilkan oleh gerakan naik turun sensor, menunjukkan banyaknya volume tetesan air yang masuk, dan sebagai counter maka pulsa tersebut digunakan untuk memberikan clock pada PIT 8253 yang dioperasikan pada mode



sensor suhu, penguat non inverting beserta rangkaian kompensasi untuk kalibrasi harga suhu. Sensor suhu yang digunakan berupa IC LM 335 yang mempunyai koefisien perubahan tegangan terhadap suhu sebesar  $10 \text{ mV}/^\circ\text{K}$ , untuk membuat koefisien tersebut menjadi lebih besar maka digunakan penguat yang mempunyai gain tertentu, disamping itu pada penguat diberi tegangan kompensasi yang berguna untuk mengoreksi pembacaan pada skala derajat celsius yang sesuai kalibrasi. Keluaran dari penguat ini siap dikonversi oleh ADC, tetapi sebelum masuk pada input ADC sinyal dilewatkan melalui pemilih (multiplexer) agar bersesuaian terhadap kanal yang akan diukur.

Penguat yang digunakan adalah penguat inverting menggunakan op-amp jenis LM 308, disini digunakan penguat inverting karena sensor suhu LM 335 dikerjakan pada ground sebagai titik referensi yang menyebabkan koefisiennya berharga negatif, dan untuk merubahnya menjadi positif perlu digunakan penguat inverting. Dari data yang ada LM 335 mempunyai koefisien dengan satuan  $\text{mV}/^\circ\text{K}$ , sedangkan untuk merubah menjadi  $\text{mV}/^\circ\text{C}$  perlu diberi suatu tegangan offset pada input positif pada op-amp. Untuk perhitungannya adalah sebagai berikut:

-Dirancang penguatan inverting dengan penguatan sebesar

$$A = -4 \text{ kali, dimana } A = -R_f/R_i.$$

-Koefisien menjadi  $40 \text{ mV}/^\circ\text{K}$ .



-Untuk  $A = -4$ , maka  $-R_f/R_i = -4$ .

-Diambil harga  $R_f = 82K$  dan  $R_i = 20,5K$  yang dibentuk dengan  $R$  sebesar  $18K$  seri dengan multiturn sebesar  $5K$ . dalam rangkaian  $R_f = R_7$ ,  $R_i = R_3$  dan  $R_8$ .

Besarnya tegangan offset setara keadaan  $0^\circ C$  atau  $273^\circ K$ , hal ini dilakukan karena suhu yang ada dirancang pada satuan  $^\circ C$ . Untuk perhitungannya adalah sebagai berikut:

-V keluaran pada LM 335 saat  $273^\circ K$  adalah  $273 * 10mV = 2,73 V$  (untuk ground sebagai referensi maka besarnya berlawanan  $= -2,73 V$ ).

-Karena  $273^\circ K$  setara  $0^\circ C$  maka output dari penguat harus menunjukkan  $0 V$ .

-Jadi besarnya tegangan offset adalah:

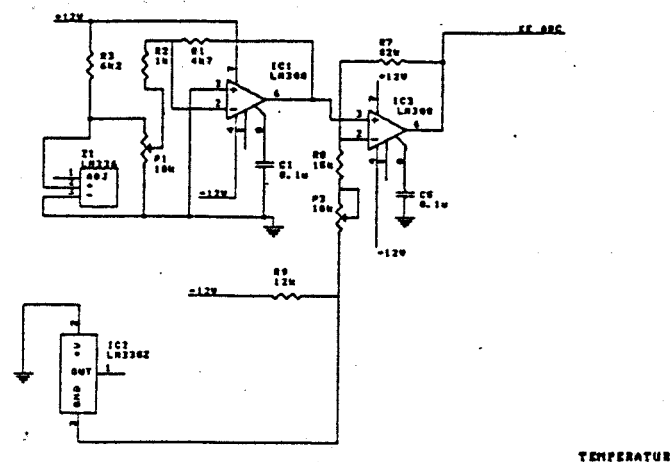
$$V_o(\text{op-amp}) = (1 + R_f/R_i)V_{\text{offset}} - R_f/R_i V_{\text{sensor}}$$

$$V_o \text{ pada saat } V_{\text{sensor}} = -2,73V \text{ sebesar } 0 V$$

$$0 = 5 V_{\text{offset}} - 4 * (-2,73)$$

$$V_{\text{offset}} = -2,18 V$$

Untuk menghasilkan tegangan offset sebesar  $-2,18 V$  dibangun dengan menggunakan op-amp yang bekerja pada penguat inverting, disini digunakan IC tegangan referensi sebagai tegangan input (LM 336  $2,5 V$ ) yang dipadukan dengan multiturn  $10 K$ , serta besar dari  $R_f = 4K7$  dan  $R_i = 1K$  (dalam rangkaian  $R_f = R_1$ ,  $R_i = R_2$ ), besar dari tahanan tersebut akan menghasilkan penguatan sebesar  $-4$  kali, sehingga untuk pengesetan tegangan sebesar  $-2,18 V$  akan dipenuhi dari rangkaian tersebut. Untuk lebih jelasnya lihat gambar 3.9.



Gambar 3.9 Rangkaian sensor suhu dengan LM 335

#### III.1.2.5. PERENCANAAN SENSOR TEKANAN DAN PENGKONDISI SINYAL

Perencanaan pada bagian ini dirancang untuk mengukur tekanan udara sekitar, bagian ini terdiri dari sensor tekanan berupa resistor jembatan, sumber arus konstan, dan penguat instrumentasi. Sensor tekanan yang dipakai adalah type SCC15A (buatan sensym) yang mampu mengukur tekanan hingga 15 PSI, karena prinsip dasar dari sensor ini adalah perubahan tahanan maka untuk dirubah kedalam tegangan perlu suatu sumber arus yang konstan sehingga keluaran tegangan yang ada murni disebabkan oleh perubahan tahanan, sensor akan mempunyai koefisien sebesar  $0,015 \text{ mV}/0,1 \text{ in-Hg}$  bila diberi sumber arus sebesar  $0,5 \text{ mA}$ , atau mempunyai koefisien tahanan  $0,03 \text{ ohm}/0,1 \text{ in-Hg}$ . Karena kecilnya koefisien yang

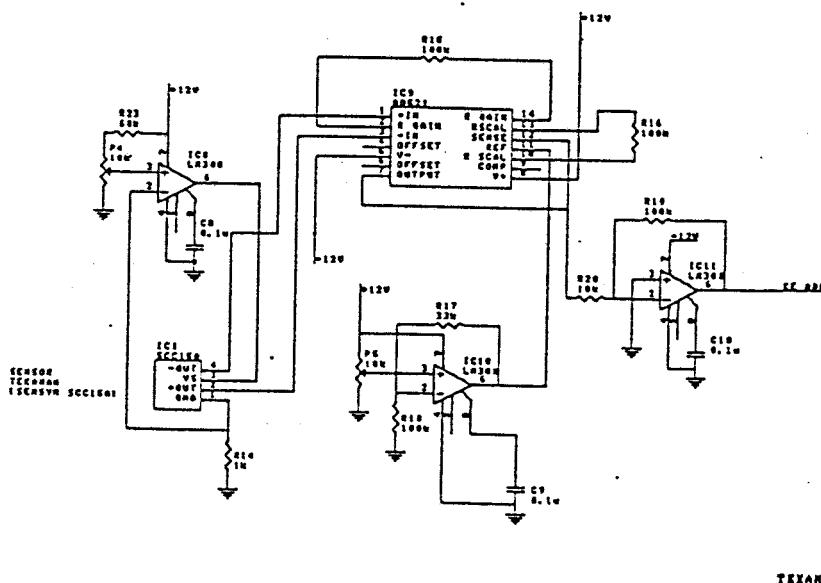
ada maka perlu dibutuhkan suatu penguat instrumentasi agar didapat keluaran tegangan yang cukup untuk diukur. Dalam rangkaian juga terdapat masukan tegangan kompensasi pada penguat instrumentasi, tegangan ini digunakan untuk offset pada saat kalibrasi, sedangkan keluaran tegangan sebelum masuk ke ADC dimasukkan kedalam pemilih agar sesuai dengan kanal yang dipakai.

Karena sensor tekanan mempunyai keluaran berupa perubahan tahanan maka untuk dikonversi dalam bentuk tegangan dibutuhkan suatu sumber arus konstan, dalam hal ini digunakan suatu op-amp yang digunakan pada mode inverting yang digunakan sebagai sumber arus konstan. Disini akan dirancang sumber arus konstan sebesar 0,5mA, maka pada multiturn P4 cukup diatur pada tegangan 0,5V dan digunakan R umpan balik sebesar 1K ( $I_{konstan} = 0,5V/1K = 0,5 \text{ mA}$ ).

Karena sinyal yang dihasilkan sangat kecil, maka perlu digunakan penguat instrumentasi. Dalam penguat ini dirancang menggunakan jenis AD521, suatu penguat instrumentasi yang pengoperasiannya hanya membutuhkan R luar yang digunakan sebagai penskala penguatan yang diinginkan (lihat pada lampiran data book).

Penguat instrumentasi ini dirancang untuk penguatan 100 kali, maka cukup digunakan R luar  $R_{15} = 100K$  dan  $R_{16} = 1K$ . Untuk bagian kompensasi dari AD521 digunakan tegangan yang dapat diatur melalui multiturn P5 dan dikuatkan oleh op-amp (LM 308) sebesar 1/3 kali, kompensasi ini digunakan

sebagai koreksi pembacaan dari keluaran saat dikalibrasi. Keluaran dari penguat instrumentasi kemudian dimasukan kedalam penguat inverting dengan penguatan sebesar -10 kali (menggunakan op-amp LM 308), sesuai dengan rumus penguatannya  $A = -R_f/R_i$ , maka  $R_f = 100K$   $R_i = 10 K$  (dalam gambar  $R_f=R_{19}$ ,  $R_i=R_{20}$ ) lihat gambar 3.10.



Gambar 3.10 Rangkaian sensor tekanan

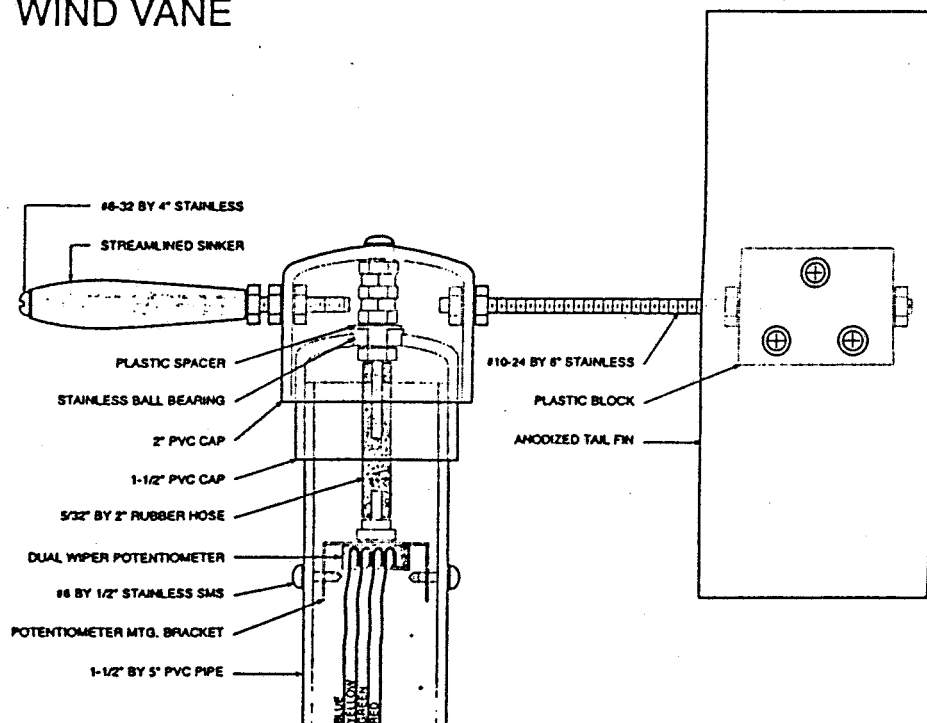
### III.1.2.6. PERENCANAAN SENSOR ARAH ANGIN DAN PENGKONDISI SINYAL

Pada perencanaan sensor arah angin dibagi menjadi dua bagian yaitu:

1. Perencanaan bagian mekanik.
2. Perencanaan bagian elektronik.

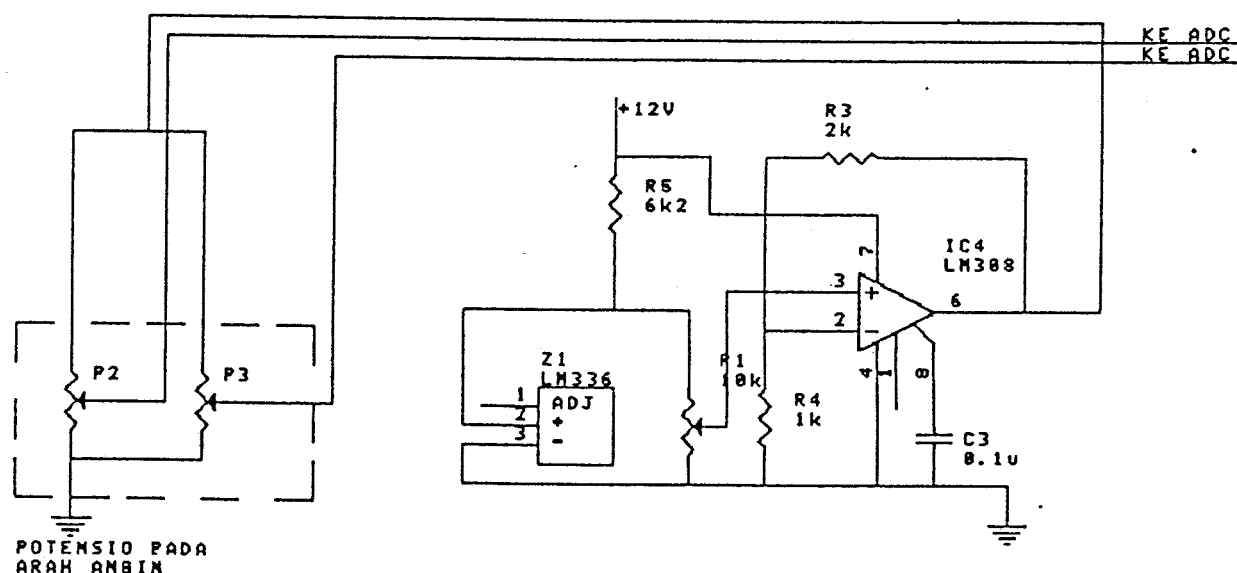
Pada bagian mekanik, dirancang suatu bentuk sirip yang bersifat labil jika tidak searah dengan arah angin, tetapi bersifat stabil bila searah dengan arah angin. Untuk maksud tersebut maka sirip tersebut harus mampu berputar sebesar sudut  $360^\circ$ , sedangkan untuk tempat penyangga dari sensor tersebut digunakan pipa PVC berdiameter 1,5". Untuk lebih jelasnya lihat gambar 3.11.

### WIND VANE



Gambar 3.11 Bagian mekanik sensor arah angin

Pada bagian elektronik, dirancang agar informasi dalam bentuk arah dapat dikonversikan dalam bentuk tegangan, didalam sensor ini digunakan suatu potensiometer yang mampu berputar sebesar sudut  $360^\circ$ . Dalam potensiometer biasa hanya mampu berputar sebesar sudut  $300^\circ$  dan terdapat daerah mati sebesar  $60^\circ$ , dalam potensiometer khusus ini untuk menghilangkan daerah mati digunakan dua wiper yang saling mengisi pada daerah matinya, prinsip kerjanya adalah bila wiper pertama telah bergerak sebesar  $300^\circ$  maka pada daerah mati langsung diisi oleh gerakan wiper kedua, sehingga potensiometer tersebut dapat bergerak secara leluasa sebesar  $360^\circ$ , tanpa mengalami daerah mati. Dari sifat tersebut maka potensiometer tersebut diberi tegangan sebesar 5,12 V (sesuai skala penuh ADC), dan informasi arah angin dapat diubah menjadi tegangan yang siap dikonversi oleh ADC.

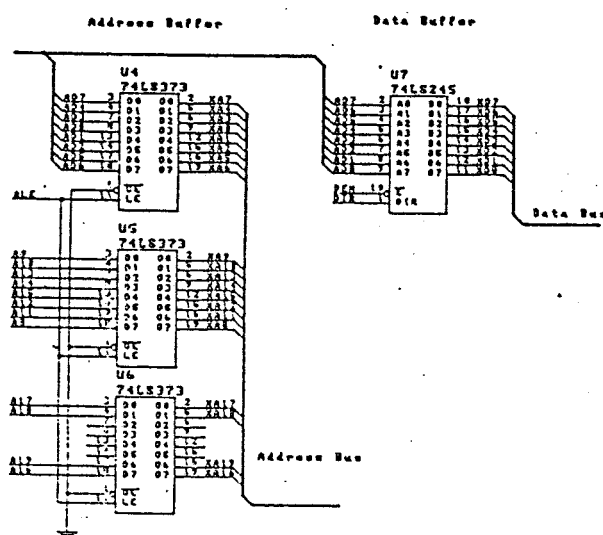


Gambar 3.12 Bagian elektronik sensor arah angin



### III.1.3.2 RANGKAIAN BUFFER ALAMAT DAN DATA

Dari uP 8088 antara alamat dan data (AD0 - AD7), bersatu dalam satu pin oleh karena itu perlu dipisah untuk membedakan keduanya, dalam hal ini digunakan IC latch 74LS373 dan IC buffer dua arah 74LS245, kedua IC tersebut dipadukan dan dikontrol oleh pin ALE dan DEN. Untuk alamat dari A8-A19 dihubungkan langsung pada IC 74LS373 yang dipadukan dengan ALE untuk memastikan bahwa yang masuk adalah informasi alamat dari uP 8088. Lihat gambar 3.14.



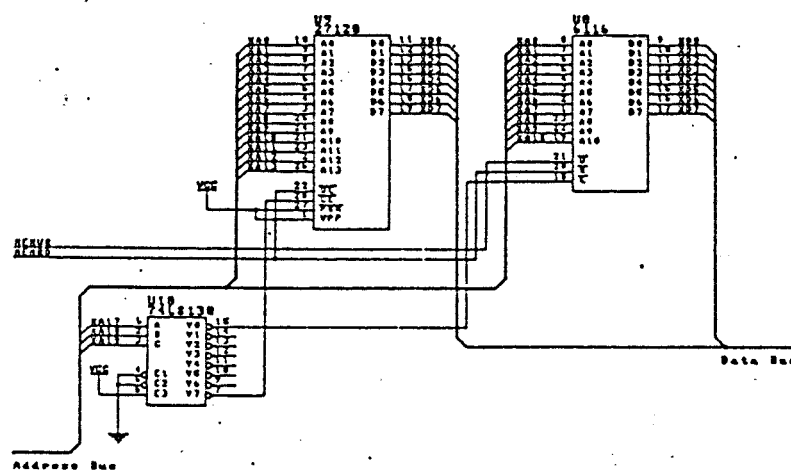
Gambar 3.14 Rangkaian buffer alamat dan data

### III.1.3.3. RANGKAIAN MEMORY

Pada sistem minimum ini digunakan memory RAM 6116 dan ROM 27128, sehingga kapasitas dari memory ini adalah RAM



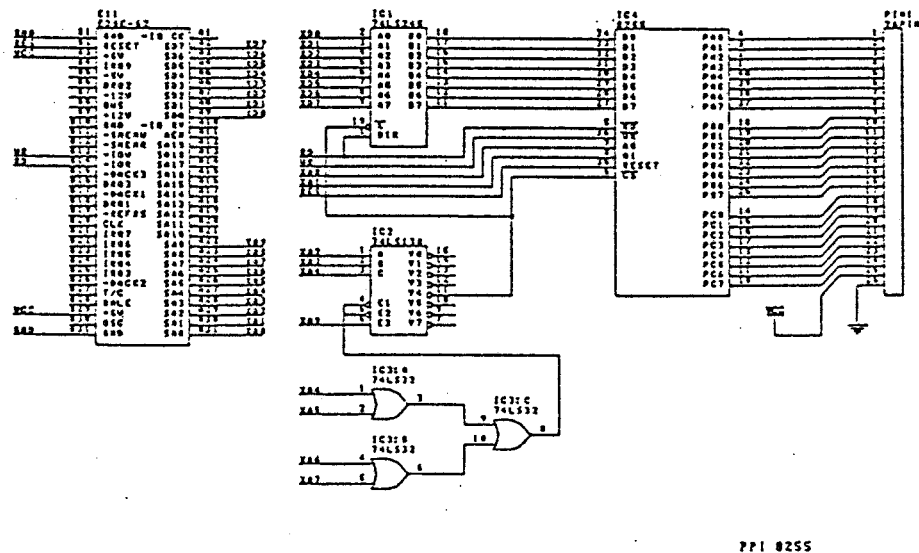
sebesar (2K x 8) ROM sebesar (16K x 8), agar mikroprosessor dapat membedakan alamat kedua memory diatas maka diperlukan suatu dekoder 74LS138, dimana peta memorynya adalah RAM 6116 terletak pada alamat 00000 - 007FF, EPROM 72128 terletak pada alamat FC000 - FFFFF. Lihat gambar 3.15.



Gambar 3.15 Rangkaian memory sistem minimum

#### III.1.3.4. RANGKAIAN I/O

Untuk rangkaian masukan dan keluaran dari sistem minimum digunakan suatu programable phireperal input/output 8255, yang diletakkan pada alamat 300h - 303h. Dalam sistem minimum ini terdapat slot pengembangan untuk menambah komponen pembantu lain yang akan dibahas selanjutnya. Lihat gambar 3.16.

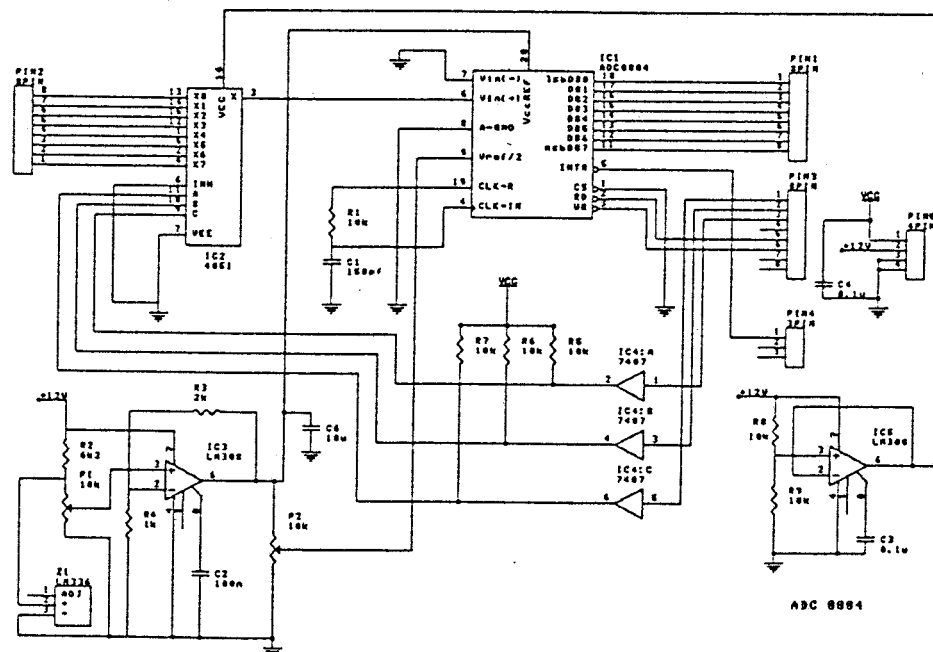


Gambar 3.16 Rangkaian PPI 8255

### III.1.4. PERANCANGAN BAGIAN ADC

Rangkaian ADC berfungsi untuk mengkonversikan data analog kedalam data digital. ADC yang digunakan dalam rangkaian ini adalah ADC 0804, ADC yang mempunyai resolusi 8 bit dan waktu konversi 100  $\mu$ det, dengan menggunakan clock yang dibangun dengan R dan C luar sebesar 10K dan 150pF (akan memberikan Fclock sebesar 600 Khz), dirancang untuk tiap bit berbobot 20 mV sehingga bila ADC menunjukkan data FF akan setara dengan input analog sebesar 5,12V.

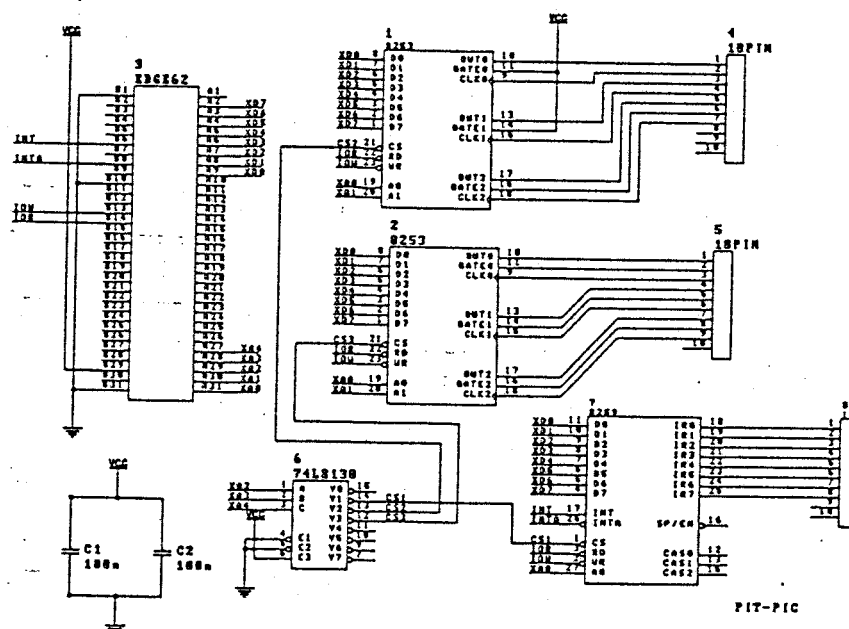
Untuk memenuhi rancangan bahwa ADC pada saat skala penuh setara dengan 5,12V maka dibutuhkan tegangan  $V_{ref}/2$  sebesar 2,56V dan  $V_{CC} = V_{ref} = 5,12V$ , maka rangkaian tegangan referensi ini dibangun dengan menggunakan Op-amp LM308 yang bekerja pada mode non inverting serta penguatan sebesar 3 kali ( $A = 1 + R_f/R_i$ ), dan digunakan IC LM336 2,5 V yang dipadukan dengan multiturn 10K sebagai penyetel pada level tegangan yang diinginkan, sedangkan untuk memilih kanal yang ada digunakan pemilih analog 4051. Untuk lebih jelas lihat gambar 3.17.



Gambar 3.17 Rangkaian ADC 0804

### III.1.5. PERANCANGAN RANGKAIAN PIT 8253 DAN PIC 8259

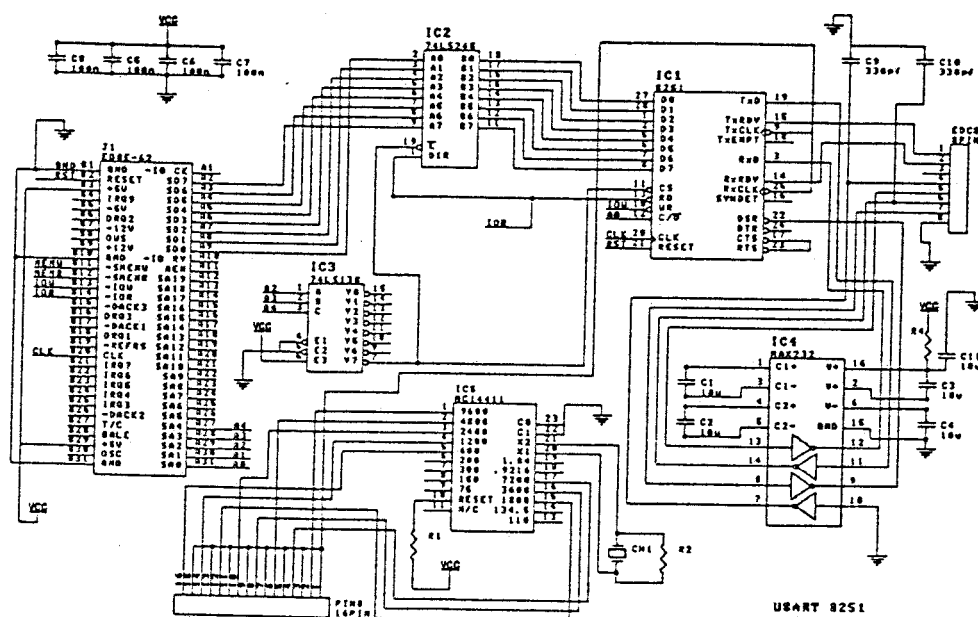
Pada sistem ini terdapat dua buah PIT dan sebuah PIC, PIT-1 terletak pada alamat 08h-0bh, PIT-2 terletak pada alamat 0ch-0fh, sedangkan PIC terletak pada alamat 04h-05h. Karena ada dua buah PIT maka terdapat 6 buah timer, dimana 2 buah dikerjakan untuk mode 1 (monostable) dan 3 buah pada mode 0 (penghitung) serta 1 buah tidak terpakai, PIT ini digunakan dalam mengukur besarnya kelembaban, kecepatan angin, dan curah hujan. Untuk PIC terletak pada alamat 04 dan 05h, dalam sistem digunakan untuk fasilitas pengambilan data secara interrupt. Lihat gambar 3.18.



Gambar 3.18 Rangkaian PIT 8253 dan PIC 8259

### III.1.6. RANGKAIAN USART 8251

Fasilitas yang disediakan untuk tampilan data pengukuran, penyimpanan data sebagai file dan sebagainya lebih mudah dilakukan oleh IBM-PC. Karena itu diadakan suatu komunikasi secara serial dengan IBM-PC. Komunikasi ini dilakukan untuk mengirim data dari minimum sistem ke dalam IBM-PC.



Gambar 3.19 Rangkaian USART 8251

Komunikasi data tersebut menggunakan standar RS232C, dimana pada komputer sudah tersedia fasilitas tersebut, sedangkan

pada minimum sistem digunakan rangkaian terpadu USART 8251. Secara lengkap rangkaian ini terdiri dari pembangkit baud rate menggunakan IC MC14411, perubah level TTL ke level RS232C digunakan MAX 232, dekoder alamat pada alamat 1ch-1dh digunakan 74LS138, buffer data digunakan 74LS245, USART itu sendiri digunakan 8251. Lihat gambar 3.19.

### III.2. PERANCANGAN PERANGKAT LUNAK

Perangkat lunak yang disimpan dalam eprom pada minimum sistem 8088 berfungsi untuk mengendalikan proses kerja perangkat keras sesuai dengan langkah-langkah yang diperlukan untuk operasi alat. Flowchart pada gambar 3.25 menunjukkan langkah-langkah yang diperlukan untuk mengoperasikan bagian perangkat keras. Langkah-langkah ini diwujudkan seperti pada program utama. Untuk mempermudah pemrograman maka program dibagi atas prosedur-prosedur untuk menjalankan tugas tertentu.

#### III.2.1. PROGRAM PADA MINIMUM SITEM

Pada minimum sitem program yang ada disimpan pada sebuah eprom, dimana dalam perangkat lunak tersebut terdapat beberapa prosedur antara lain:

##### PROSEDUR INISIALISASI

Prosedur ini berfungsi menyiapkan kerja perangkat keras, seperti melokasi memory, menentukan segment memory, menentukan operasi PPI 8255, PIC 8259, PIT 8253, USART 8251,

mempersiapkan interupt untuk IRO. Prosedur ini hanya dilakukan sekali saja.

#### PROSEDUR KANAL

Prosedur ini berfungsi untuk menentukan kanal yang akan diambil datanya, dalam ini jumlah kanal adalah 6 macam sesuai dengan jumlah parameter yang akan diukur, jadi tidak terjadi urutan yang acak akan pengambilan data dari masing-masing parameter.

#### PROSEDUR KONVERSI

Prosedur ini berguna untuk melakukan konversi pada bagian ADC 0804. Dalam hal ini parameter yang mengalami konversi adalah parameter suhu, arah angin, dan tekanan.

#### PROSEDUR BCD

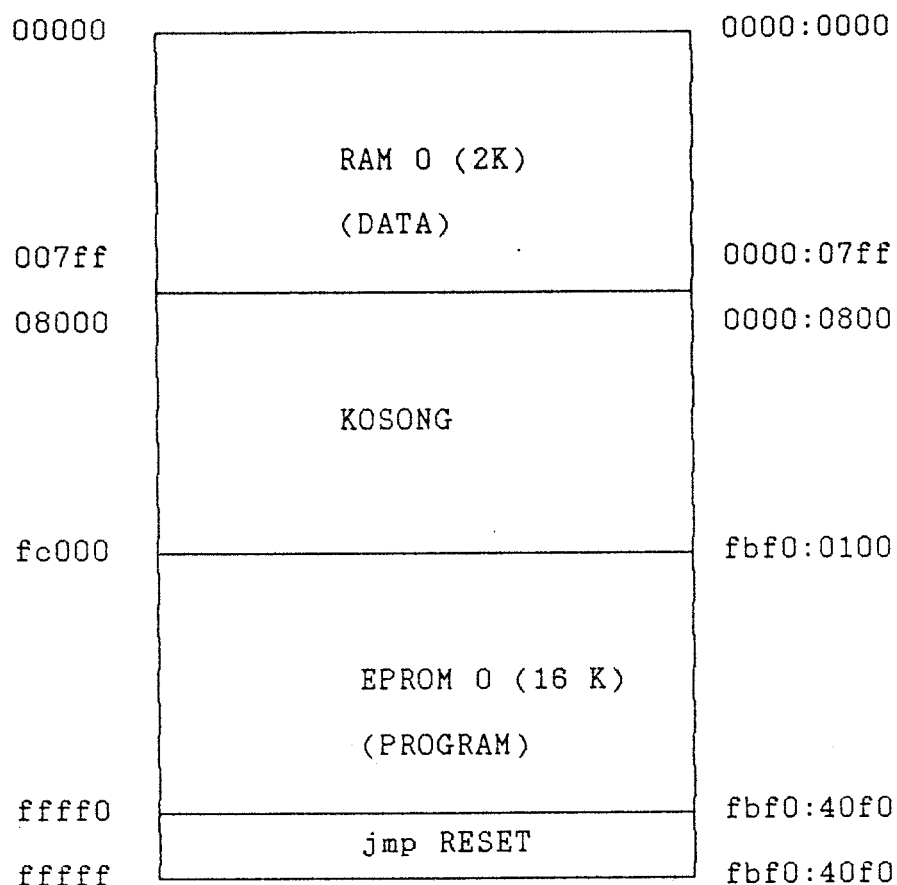
Prosedur ini berguna untuk melakukan perubah bilangan biner menjadi bilangan BCD, yang kemudian disimpan pada memory RAM untuk digunakan selanjutnya.

#### PROSEDUR KIRIM

Prosedur ini digunakan untuk melakukan proses kirim data kedalam komputer melalui RS232C, data yang dikirim berupa informasi dari keenam parameter yang disimpan dalam RAM dalam satu deret.

### III.2.2. PETA ALAMAT SISTEM

Untuk lebih jelas kita lihat peta memory dari sistem:



### III.2.3. PROGRAM PASCAL PADA IBM PC

Program pascal yang terletak pada komputer merupakan program bantu untuk tampilan, menyimpan data dalam file, interval pengambilan data, cetak data dan program bantu lainnya. Pada program terdapat beberapa prosedur antara lain:

#### PROCEDUR INITIALISASI

Procedur ini berisi segala inisialisasi yang berhubungan pada program.

#### PROCEDUR MENU UTAMA

Prosedur ini berisi tentang menu yang ada pada tampilan



yaitu: monitor data, interval waktu, nama data.

#### PROCEDUR INTERVAL

Prosedur ini berisi pengambilan waktu sampling dari data yang akan diukur.

#### PROCEDUR RS-232

Prosedur ini berisi program pengambilan data melalui fasilitas RS-232.

#### PROCEDUR PENGAMATAN

Prosedur ini digunakan untuk menampilkan data pada monitor pada bentuk animasi.

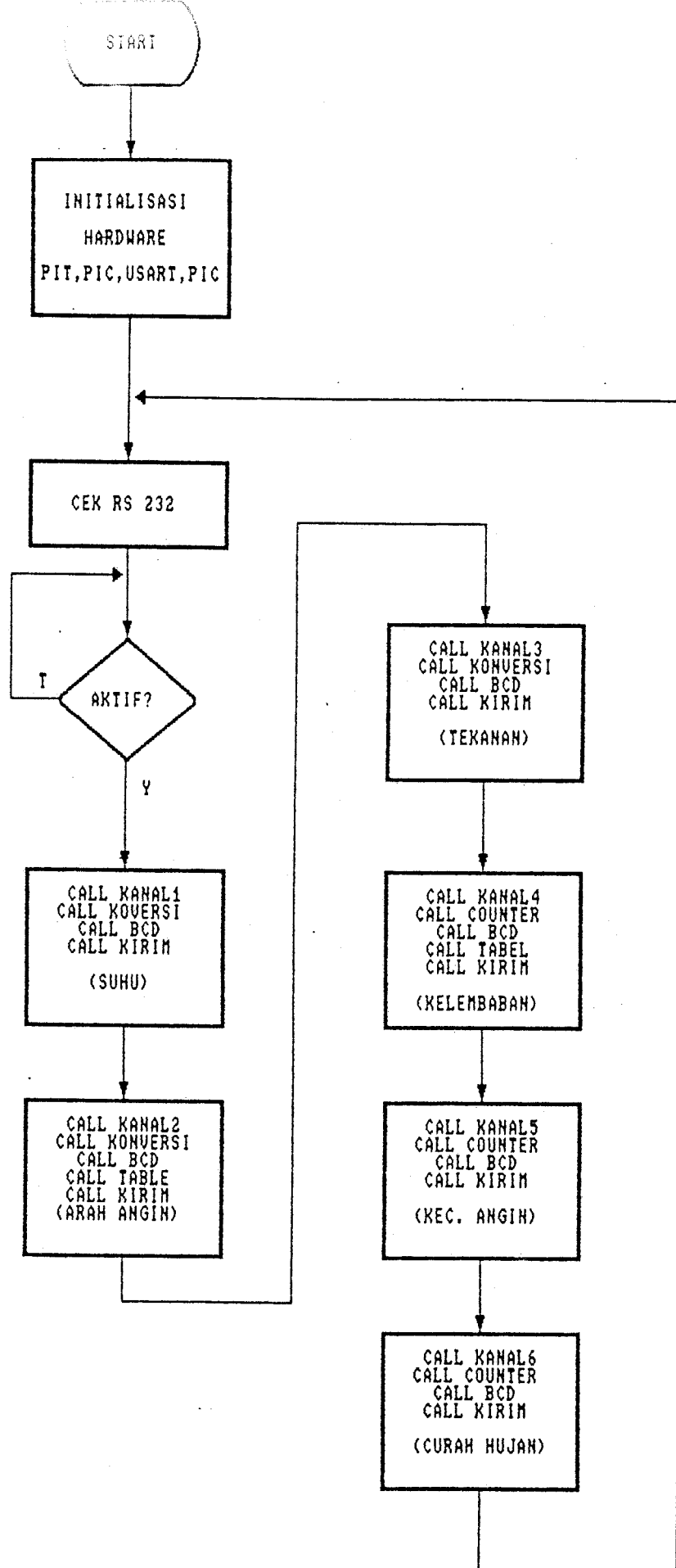
#### PROCEDUR SIMPAN

Prosedur ini berguna untuk melakukan penyimpanan data file dengan selang waktu tertentu.

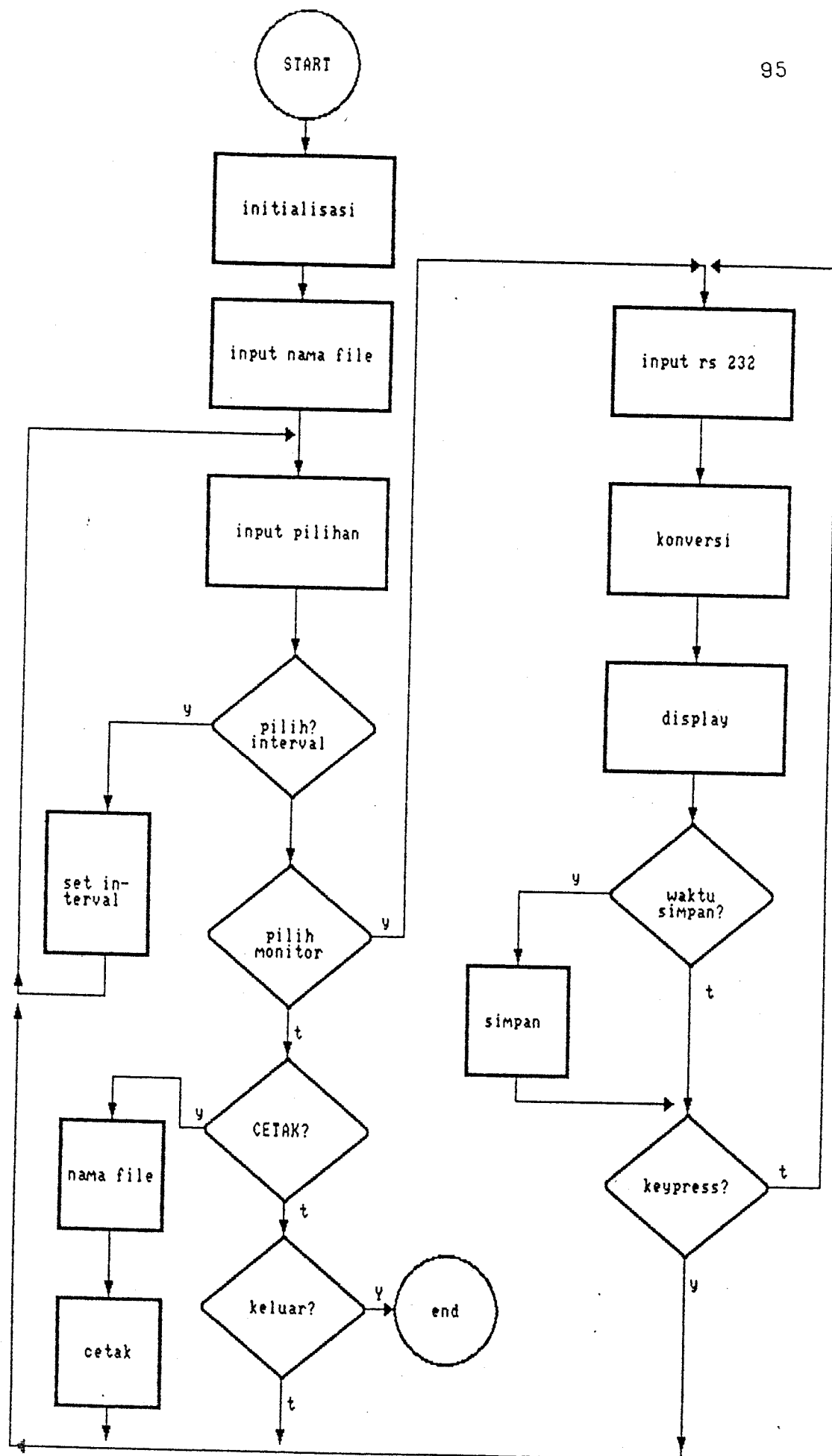
#### PROCEDUR CETAK

Prosedur ini berguna untuk melakukan percetakan pada printer.

Untuk lebih jelasnya lihat flowchart yang ada.



FLOWCHART PROGRAM PADA MINIMUM SISTEM



FLOWCHART PROGRAM TAMPILAN

## BAB IV

### PENGUKURAN DAN LINIERISASI

Untuk memperoleh hasil pengukuran yang akurat, sensor dari sistem perlu dikalibrasi dan dilinierisasi dengan menggunakan perangkat lunak.

#### IV.1. BAGIAN ADC SISTEM

Pada bagian ini dilakukan pengukuran dengan memberikan input tegangan tertentu pada ADC untuk dibaca bobotnya.

Tabel 5.1. Bobot data pada ADC

Tegangan input (mV)	Bobot ADC
0	0
100	5
200	10
300	15
400	20
500	25
600	30
700	35
800	40
900	45
1000	50
1100	54
1200	59
1300	65
1400	70
1500	75
1600	80
1700	85
1800	90
1900	95

2000	100
2100	105
2200	109
2300	114
2400	119
2500	126
2600	129
2700	136
2800	140
2900	145
3000	150
3100	155
3200	160
3300	165
3400	169
3500	175
3600	180
3700	185
3800	191
3900	196
4000	200
4100	205
4200	209
4300	215
4400	219
4500	224
4600	230
4700	235
4800	240
4900	245
5000	250
5100	255

Dengan menggunakan software maka didapat persamaan regresi liniernya adalah:

$$Y = -0.10859 + 0.049996.X$$

Dimana X = input tegangan (mV)

Y = bobot ADC

#### IV.2. BAGIAN SENSOR KELEMBABAN

Pada bagian ini dilakukan pengukuran antara besarnya frekwensi yang keluar dengan kelembaban yang ada, teknik pengukuran dilakukan pada udara bebas, dimana dari hasil survey ada Badan Meteorologi Juanda Surabaya didapat rentang kelembaban antara 50% - 80%. Dalam pengukuran ini digunakan hygrometer Hisamatsu.

Tabel 5.2 Pengukuran kelembaban terhadap frekwensi

Kelembaban (%)	Frekwensi output (Hz)
50	955
53	952
55	949
58	942
60	938
62	933
66	923
67	917
70	913
72	907
74	903
75	900
77	894
78	892
80	879

Dengan bantuan software maka didapat persamaan regresi liniernya adalah:

$$Y = 434.1197 + (-0.39970).X$$

Dimana X = frekwensi output (Hz)

Y = kelembaban (%)

#### IV.3 BAGIAN SENSOR CURAH HUJAN

Pada bagian ini pengukuran dilakukan dengan menuangkan air pada volume tertentu, kemudian kita lihat penunjukan besarnya hitungan pada sistem.

Tabel 5.3 Pengukuran curah hujan terhadap hitungan sistem

Curah hujan (mm)	Hitungan
10	10
20	21
30	33
40	45
50	56
60	68
70	80
80	92
90	103
100	115
110	127
120	139
130	150
140	162
150	174
160	186
170	197
180	210
190	221
200	232
210	244
220	255
230	267
240	278
250	291
260	303
270	314
280	325
290	337
300	349

Dengan bantuan software maka didapat persamaan regresi

liniernya adalah

$$Y = 1.626069 + 0.854609 \cdot X$$

Dimana X = hitungan pada sistem

Y = curah hujan yang sebenarnya

#### IV.4 BAGIAN SENSOR TEMPERATUR

Pada bagian ini digunakan pemanas untuk memanaskan sensor dan termometer pembanding.

Tabel 5.4 Pengukuran temperatur terhadap bobot ADC

Temperatur (°C)	Bobot ADC
26	52
30	59
35	73
40	82
45	90
50	100
55	112
60	120
65	133
70	142

Dengan bantuan software maka didapat persamaan regresi linier adalah:

$$Y = -0.96783 + 0.508563 \cdot X$$

Dimana X = bobot ADC

Y = suhu udara (°C)



#### IV.5 BAGIAN SENSOR TEKANAN

Pada bagian ini pengukuran didapat dari tabel spesifikasi dari komponen sensor, sedangkan untuk tekanan udara pada tempat setempat dilakukan survey pada Station Meteorologi Juanda Surabaya. Tekanan udara relatif tetap karena faktor yang berperan adalah ketinggian pengukuran dari permukaan laut, di daerah Surabaya tekanan berkisar 1014-1015 mBar, pada ketinggian 2.6 M dari permukaan laut.

Tabel 5.5 Pengukuran tekanan terhadap bobot ADC

Tekanan (in-Hg)	Bobot ADC
27.3	0
27.4	7
27.5	12
27.6	20
27.7	27
27.8	33
27.9	40
28.0	47
28.1	54
28.2	60
28.3	67
28.4	74
28.5	80
28.6	88
28.7	94
28.8	100
28.9	109
29.0	110
29.1	121
29.2	128
29.3	134
29.4	143
29.5	148
29.6	155
29.7	161
29.8	167

29.9	175
30.0	182
30.1	188
30.2	194
30.3	202
30.4	209
30.5	215
30.6	222
30.7	229
30.8	235
30.9	242
31.0	249
31.1	255

Dengan bantuan software didapat persamaan regresi liniernya adalah

$$Y = 27.30368 + 0.014859.X$$

Dimana X = bobot ADC

Y = tekanan yang sebenarnya (in-Hg)

#### IV.6. BAGIAN SENSOR ARAH ANGIN

Pada bagian ini diukur antara penunjukan sirip (dalam derajat) dengan bobot ADC pada sistem.

Tabel 5.6 Pengukuran sudut sirip terhadap bobot ADC

Sudut sirip (°)	Bobot ADC
0(360)	9
10(370)	2
20	373
30	359
40	348



50	340
60	326
70	316
80	308
90	398
100	285
110	277
120	266
130	254
140	245
150	232
160	225
170	214
180	202
190	190
200	178
210	167
220	155
230	145
240	133
250	121
260	113
270	103
280	95
290	87
300	78
310	67
320	53
330	40
340	31
350	20
360	9

Dengan menggunakan software maka didapat persamaan regresi liniernya adalah

$$Y = -0.941575.X + 369.061$$

Dimana X = bobot ADC

Y = arah angin (derajat)

#### IV.7 BAGIAN SENSOR KECEPATAN ANGIN

Pada bagian ini dilakukan pengukuran kecepatan angin terhadap frekwensi yang dihasilkan, sebagai standar digunakan anemometer Daiwa.

Tabel 5.7 Pengukuran kecepatan angin terhadap frekwensi

Kecepatan angin (m/s)	Frekwensi output (Hz)
0	0
3.5	4
5	7
6	10
7	12
7,75	14

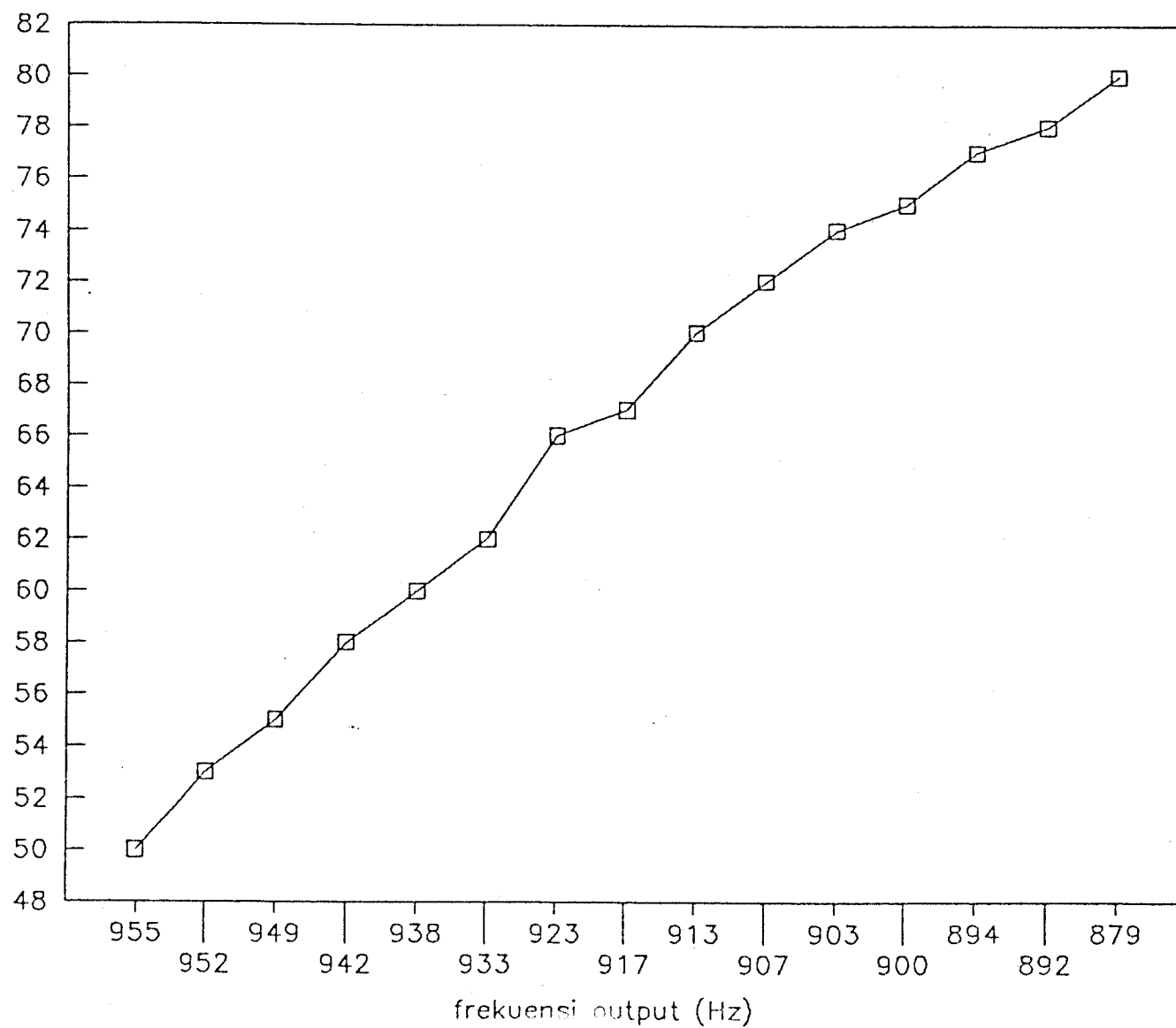
Dengan menggunakan software maka didapat persamaan regresi tak liniernya adalah

$$Y = 1,4644.X^{0,626492}$$

Dimana X = frekwensi sistem

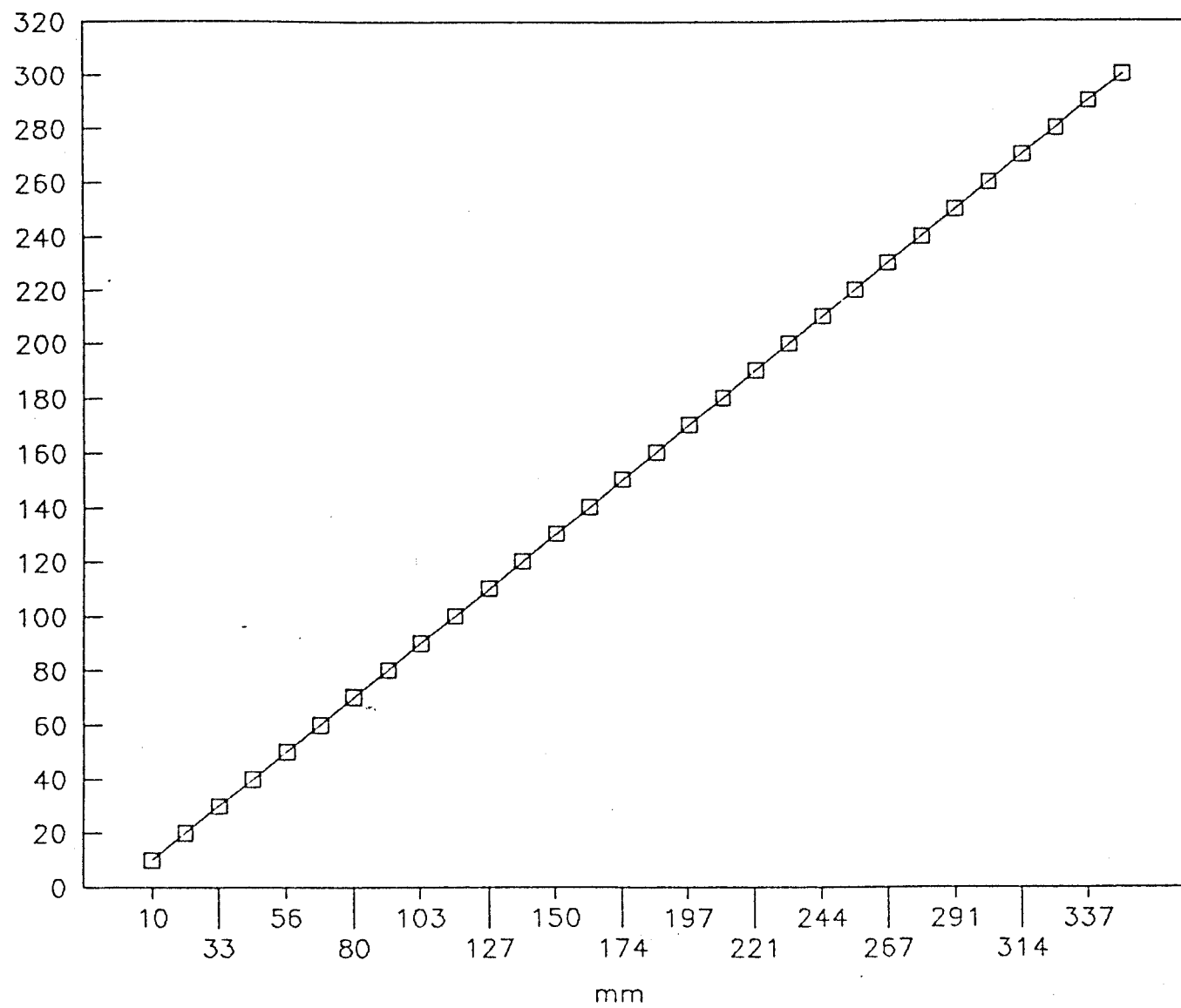
Y = kecepatan angin (m/s)

## KELEMBABAN

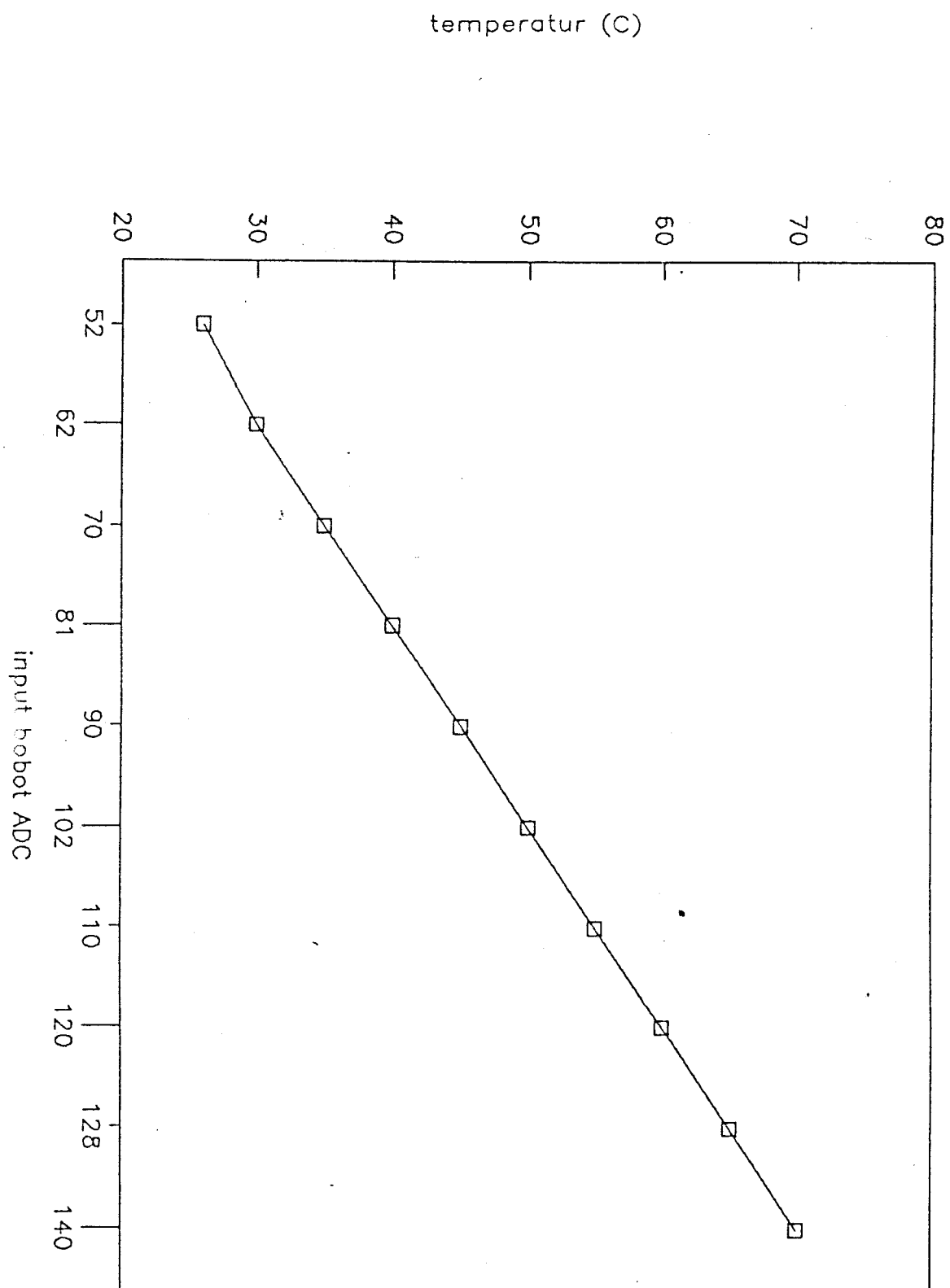


## HUJAN

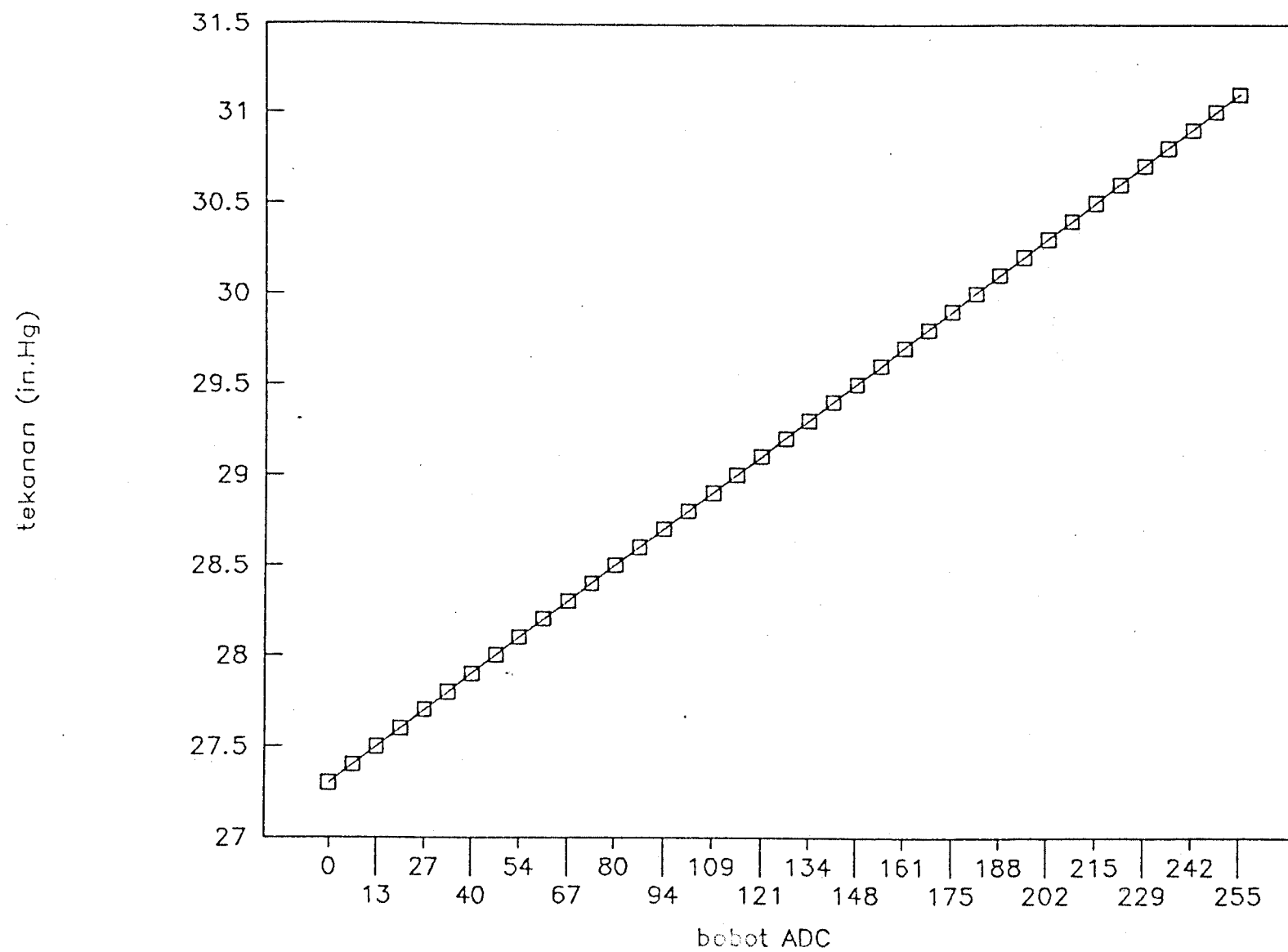
counter sensor



## SUHU



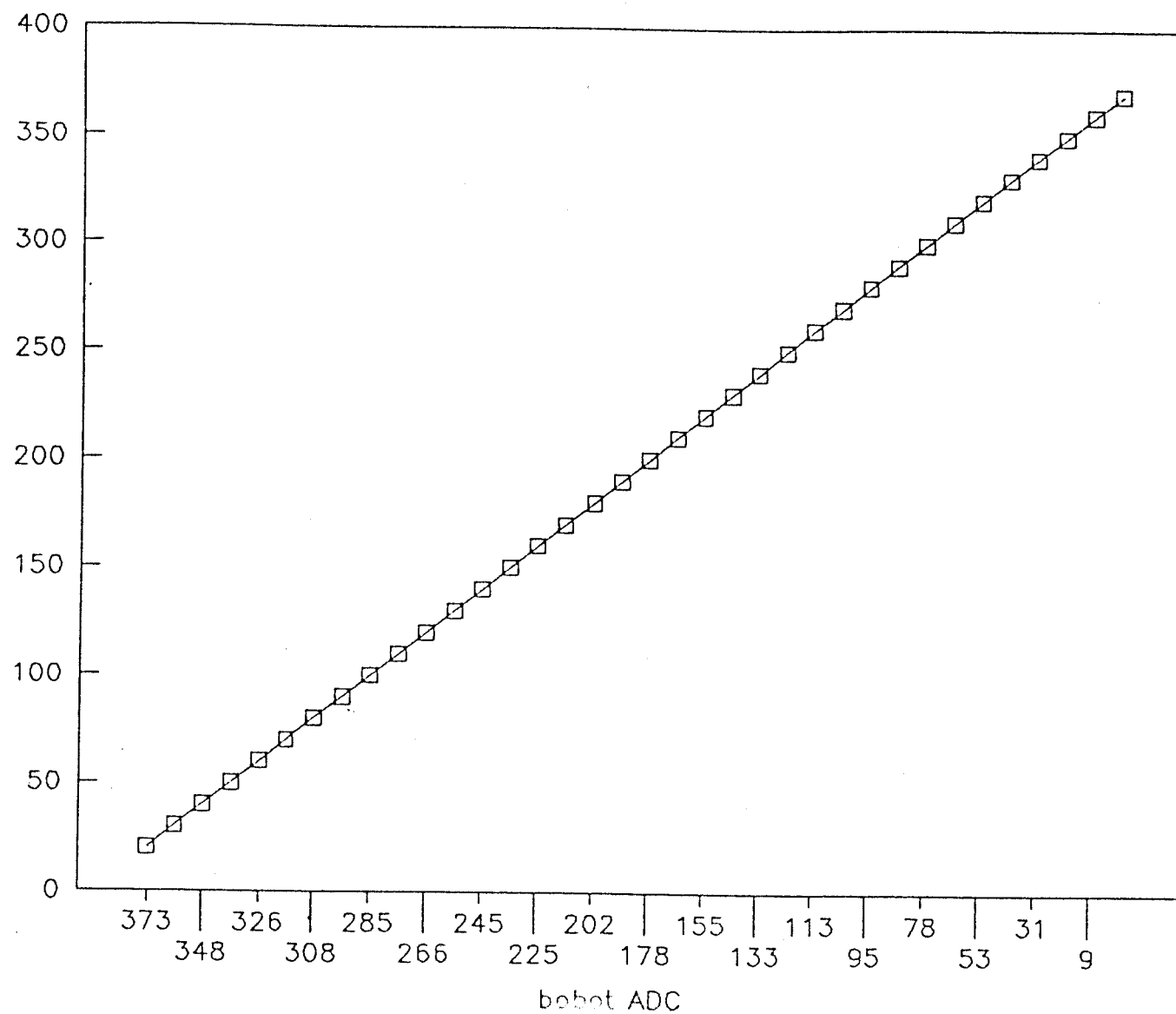
## TEKANAN



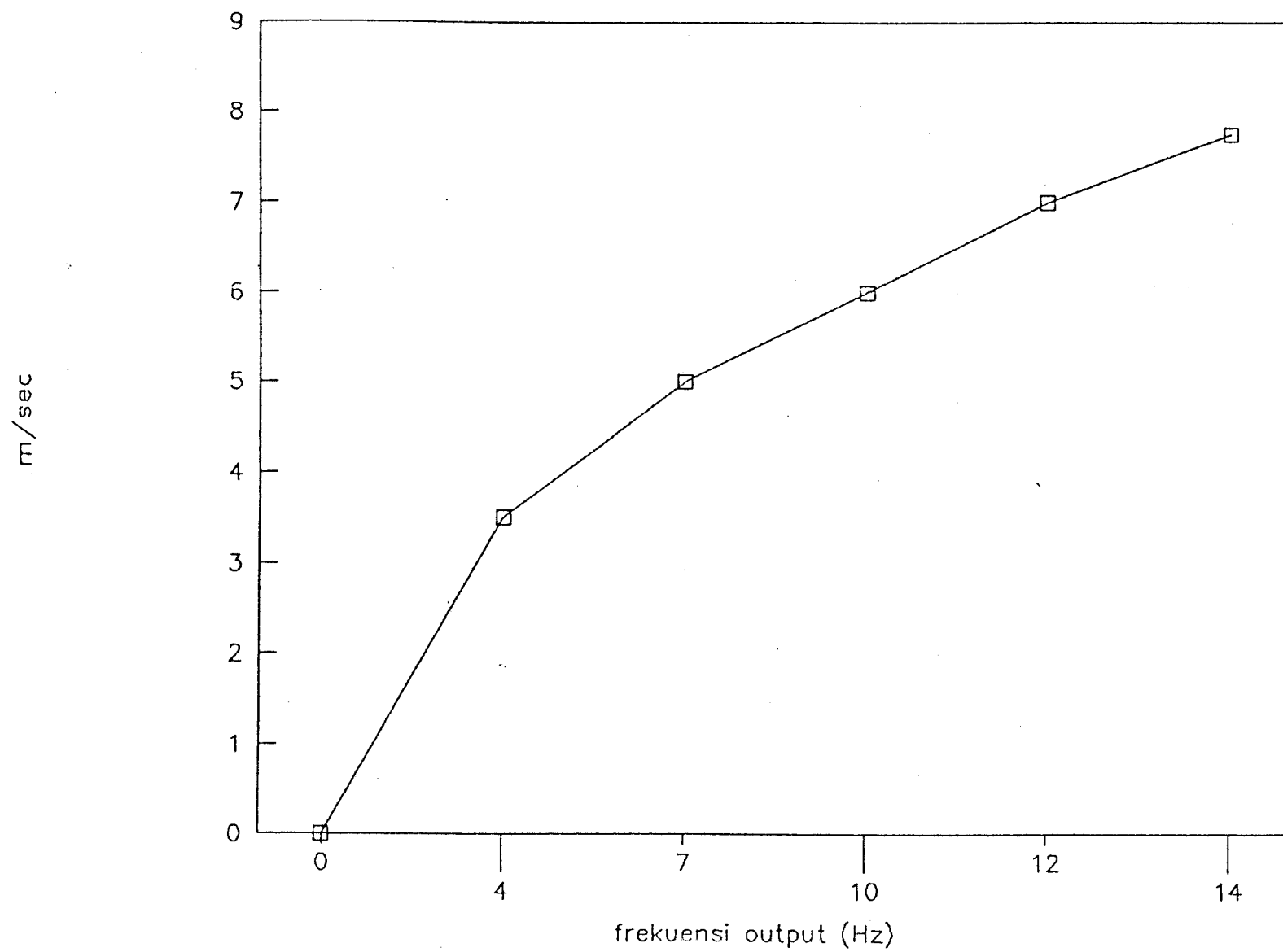


## ARAH ANGIN

derajat

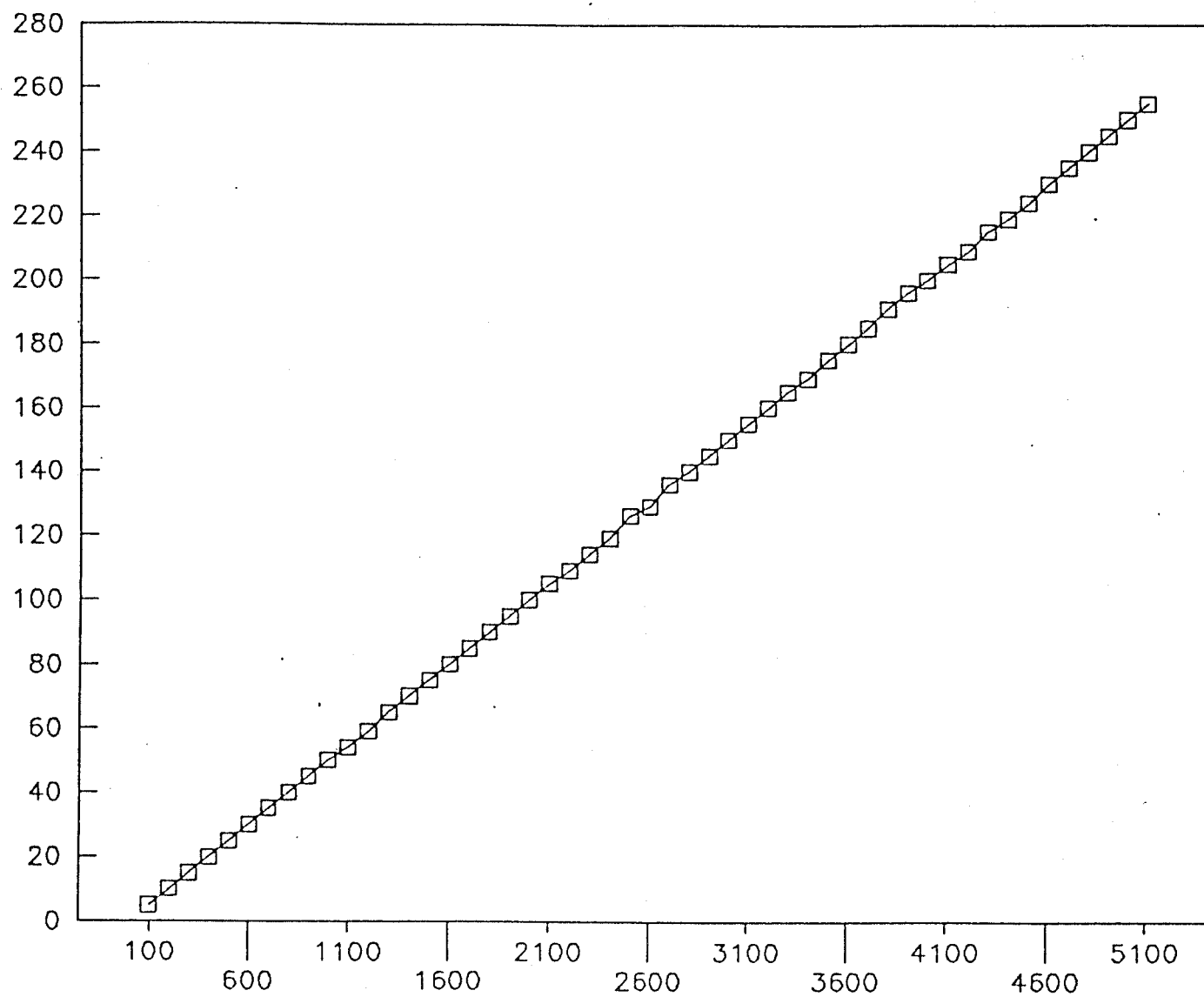


## KECEPATAN ANGIN



# ADC

111



## BAB V

### PENUTUP

#### V.1. KESIMPULAN

Dari pengamatan selama analisa untuk perencanaan, disain perangkat keras maupun perangkat lunak, pembuatan, pengukuran dan ujicoba, dapat diambil beberapa kesimpulan:

1. Penggunaan sistem minimum 8088 yang digunakan untuk akuisisi data mempunyai banyak keuntungan antara lain:
  - Keluwesan pada perangkat lunak.
  - Kecepatan proses.
  - Untuk sistem pengembangan lebih lanjut.
2. Kombinasi 6 parameter seperti suhu, tekanan, kecepatan angin, arah angin, kelembaban, curah hujan merupakan syarat minimum untuk akuisisi data meteorologi.
3. Penggunaan fasilitas RS 232 memungkinkan jarak pengiriman data yang relatif jauh (antara letak sensor dan IBM-PC).
4. Penggunaan IBM PC sangat membantu dalam penampilan data, penyimpanan data, interval pengambilan data.
5. Penggunaan sensor kapasitansi untuk mengukur kelembaban yang dikombinasikan dengan astable multivibrator sangat effisien, bila digunakan sensor jenis lain.
6. Sensor tekanan berupa piezo resistive sangat cocok digunakan sebagai pengukur tekanan rendah (tekanan udara sekitar).
7. Sensor curah hujan digunakan model bucket karena dianggap

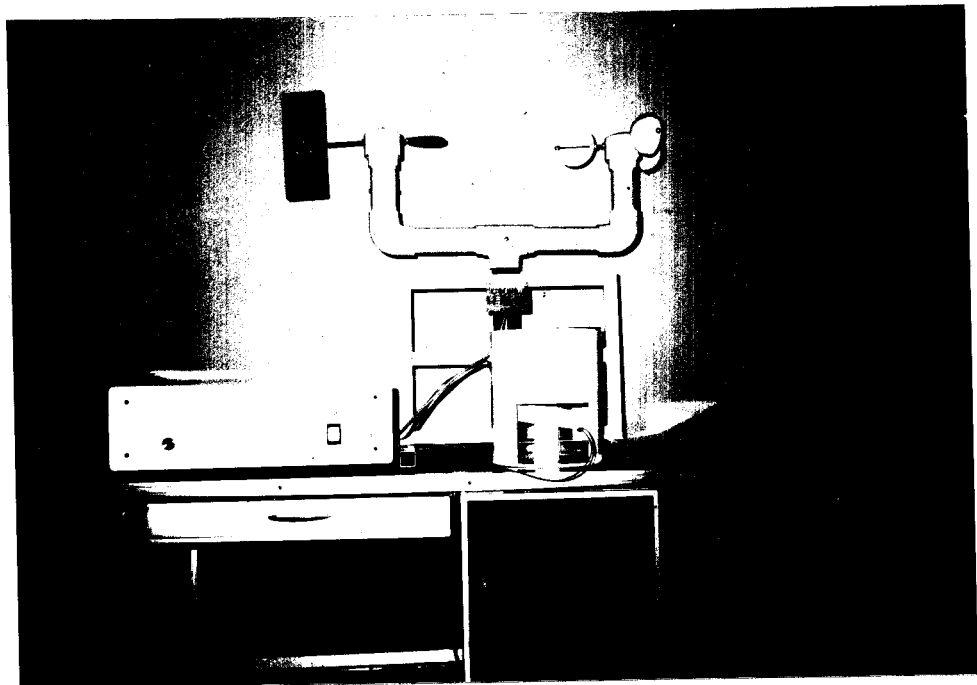
paling teliti dibanding model lain.

8. Sensor suhu digunakan jenis sensor solid state (IC) karena mempunyai respon dan karakteristik yang linier pada suhu.
9. Penggunaan sensor kecepatan angin menggunakan model mangkok sedangkan sensor arah angin digunakan dengan potensiometer yang mempunyai ketelitian hingga  $1^\circ$ .

## DAFTAR PUSTAKA

1. Coughlin, Robert F., Frederick F. Driscoll, Herman W. (penerjemah), PENGUAT OPERASIONAL DAN RANGKAIAN TERPADU LINIER, Erlangga, Jakarta, 1985
2. Charles A. Schuler. William L. Mc Namee, INDUSTRIAL ELECTRONIC AND ROBOTIC, Mc Graw-Hill, Singapore, 1968.
3. Gayakwad, Ramakant A., Op-AMP & LINIER INTEGRATED CIRCUITS, Prentice Hall of India Private Limited, New Delhi, 1988.
4. Hall, Douglas V., MICROPROCESSOR AND INTERFACING: Programming and Hardware, McGraw-hill Inc, Singapore, 1986.
5. Hartono Partoharsojdo, TUNTUNAN PRAKTIS PEMROGRAMAN BAHASA ASSEMBLY, PT. Elex Media Komputindo, Jakarta, 1990.
6. Hogenboom, P., DATA SHEET BOOK 3 Catatan Aplikasi, PT. Elex Komputindo Kelompok Gramedia, Jakarta, 1989.
7. Malloney, Timothy J., INDUSTRIAL SOLID STATE ELECTRONICS DEVICES AND SYSTEMS, 2nd Edition, Prentice Hall, Inc, Englewood Cliffs, New Jersey 0762.
8. Uffenback, John, THE 8086/8088 FAMILY DESIGN PROGRAMING AND INTERFACING, Prentice-Hall Inc., 1988.
9. -----, LINIER DATA BOOK, National Semiconductor Corp., Santa Clara, 1982.
10. -----, INTEL COMPONENT DATA CATALOG, Intel corp., Literature Distribution, Santa Clara, 1982

BENTUK SISTEM YANG TELAH DIBUAT



```
;Dadang Samuel Nrp. 2912201796
;program akuisisi 6 parameter Meteorologi
;inisialisasi ppi,pic,usart
```

```
;-----
```

```
;PPI-1
```

```
pa1 equ 300h
```

```
pb1 equ 301h
```

```
pc1 equ 302h
```

```
cw1 equ 303h
```

```
;PPI-2
```

```
pa2 equ 308h
```

```
pb2 equ 309h
```

```
pc2 equ 30ah
```

```
cw2 equ 30bh
```

```
;PIC
```

```
pic0 equ 04h
```

```
pic1 equ 05h
```

```
;PIT-1
```

```
countA0 equ 08h
```

```
countA1 equ 09h
```

```
countA2 equ 0ah
```

```
contregA equ 0bh
```

```
;PIT-2
```

```
countB0 equ 0ch
```

```
countB1 equ 0dh
```

```
countB2 equ 0eh
```

```
contregB equ 0fh
```

```
;USART
```

```
datareg equ 1ch
```

```
contreg equ 1dh
```

```
;led
```

```
led1 equ 10h
```

```
led2 equ 14h
```

```
org 100h
```

```
type_64 label word
```

```
org 114h
```

```
buff label byte
```

```
;-----
```

```
;program utama
```

```
;-----
```

```
org 100h
```

```
start: jmp utama
```



UTAMA:

```
cli          ;inisialisasi 8088
xor ax,ax
mov ss,ax
mov es,ax
mov ds,ax
xor si,si
xor di,di
mov sp,7ffh

mov ax,offset int_64    ;interrupt vektor type 64
mov es:[type_64],ax
mov es:[type_64+2],cs

mov al,00010011b        ;ICW1
mov dx,pic0
out dx,al

mov al,01000000b        ;ICW2
mov dx,pic1
out dx,al

mov al,00000001b        ;ICW4
out dx,al
mov al,11111110b        ;OCW1
out dx,al

    mov dx,contreg      ;inisialisasi 8251
    mov al,00h          ;seperti di V.Hall
    out dx,al
    mov cx,200
d0:  loop d0
    out dx,al
    mov cx,200
d1:  loop d1
    out dx,al
    mov cx,200
d2:  loop d2
    mov al,40h
    out dx,al
    mov cx,200
d3:  loop d3
    mov al,01001101b
    out dx,al
    mov cx,200
d4:  loop d4
    mov al,00110111b
    out dx,al
```

```

mov al,01110011b      ;inisialisasi PIT-B
mov dx,contregB
out dx,al

mov dx,countB1         ;load lsb counter-B1
mov al,00h
out dx,al

mov al,02h            ;load msb counter-B1
out dx,al

mov al,00110011b      ;inisialisasi PIT-B0
mov dx,contregB
out dx,al

mov dx,countB0         ;load lsb counter-B0
mov al,00h
out dx,al

mov al,04h            ;load msb counter-B0
out dx,al

mov al,80h            ;inisialisasi PPI-1
mov dx,cw1
out dx,al

mov dx,cw2            ;inisialisasi PPI-2
mov al,99h
out dx,al

mov dx,contregA        ;inisialisasi PIT-A
mov al,00110000b
out dx,al

mov al,0ffh           ;load lsb counter-A0
mov dx,countA0
out dx,al
mov al,0ffh           ;load msb counter-A0
out dx,al

mov dx,pci            ;kirim trigger pd PIT
mov al,00h            ;digunakan PC1
out dx,al
mov al,01h
out dx,al
mov cx,000fh
delayb: loop delayb
mov al,00h
out dx,al

mov dx,contregB        ;inisialisasi PIT-B
mov al,10110000b
out dx,al

```

```

mov al,0ffh          ;load lsb counter-B2
mov dx,countB2
out dx,al
mov al,0ffh          ;load msb counter-B2
out dx,al

mov cx,0018h          ;jumlah memory data
mov bx,0000h
lea di,buff

kosong: mov es:[di+bx],00 ;kosongkan data
        inc bx
        loop kosong

        awal: mov bx,0000          ;keadaan awal bx

awala: push bx
        mov bh,01h          ;kanal ADC untuk arah angin-1
        call konversi          ;konversi

cmp al,35h          ;53
jb awall
cmp al,00fch          ;252
ja awall
call konversi
sub al,35h
jmp ubah

awall:          ;kanal ADC untuk arah angin-0
        mov bh,00h
        call konversi          ;konversikan
        mov ah,00h
        add ax,008ch          ;140
ubah: mov bx,00010h          ;input parameter pada BCD
        call bcd          ;ubah BCD

mov bh,00000010b          ;kanal untuk suhu
call konversi          ;konversi

xor ah,ah
mov bx,0014h          ;input parameter pada BCD
call bcd          ;ubah BCD

mov bh,00000011b          ;kanal untuk tekanan
call konversi          ;konversi

xor ah,ah
mov bx,000ch          ;input parameter pada BCD
call bcd          ;ubah BCD

call huay          ;kelembaban
call hujan          ;curah hujan

```

```

sinil: mov dx,contreg
dsr1: in al,dx
      and al,10000101b ;cek status USART
      cmp al,10000101b
      jne dsr1         ;berhasil?
      mov dx,datareg
      mov al,05h       ;kirim header
      out dx,al        ;masuk serial

      mov cl,00h
      mov bx,0017h     ;counter untuk jumlah data
      sti

sini:  mov dx,contreg
dsr:   in al,dx
      and al,10000101b ;cek status USART
      cmp al,10000101b
      jne dsr          ;berhasil?
      call kirim        ;kirim data
      cmp cl,00h        ;cek harga cl dari "return
                        ;value" pada procedure kirim

      jne sini
      pop bx
      inc bh
      cmp bh,02h        ;tunggu 2 kali kirim
      jne maneh

      mov dx,pcl        ;kirim interupt type-64
      mov al,00h
      out dx,al
      mov al,04h
      out dx,al
      mov cx,000fh
      tunda1: loop tunda1
      mov al,00h
      out dx,al

      jmp awal          ;ulang ke "awal"

maneh: jmp awala        ;ulang ke "awala"

delay proc near         ;procedure delay
push cx
mov cx,0ffffh
kene: loop kene
pop cx
ret
delay endp

```

```

;-----
; kirim proc near      ;procedure untuk kirim data
    lea di, buff
    mov al, es:[di+bx]
    add al, 30h
    mov dx, datareg
    out dx, al
    dec bx
    mov cl, bl
    add cl, 1
    ret
kirim endp
;-----

```

```

;-----
; huay proc near      ;procedur untuk kelembaban
    mov dx, contregA
    mov al, 01110000b
    out dx, al
    mov al, 0ffh
    mov dx, countA1
    out dx, al
    mov al, 0ffh
    out dx, al

```

```

    mov dx, pci
    mov al, 00h
    out dx, al
    mov al, 02h
    out dx, al
    mov cx, 000fh
delay1: loop delay1
    mov al, 00h
    out dx, al

```

```

    mov cx, 0004h
herel: push cx
    call delay
    pop cx
    loop herel
    mov cx, 2000h
delay2 loop delay2

```

```

    mov al, 01000000b
    mov dx, contregA
    out dx, al
    mov dx, countA1
    in al, dx
    mov ah, al
    in al, dx
    xchg ah, al
    mov dx, 0ffffh
    sub dx, ax
    mov ax, dx
    mov bx, 0008h
    call bcd

```

```

    ret
huay endp
;-----

```

```

;-----
hujan proc near      ;procedure untuk curah hujan

mov al,10000000b
mov dx,contreg8
out dx,al
mov dx,countB2
in al,dx
mov ah,al
in al,dx
xchg ah,al
mov dx,0ffffh
sub dx,ax
mov ax,dx

mov bx,0000h
call bcd
ret
hujan endp
;-----

;-----
konversi proc near   ;procedure untuk konversi ADC
mov al,00110000b
add al,bh
mov dx,pb2
out dx,al

mov al,00010000b
add al,bh
out dx,al

mov al,00110000b
add al,bh
out dx,al

mov dx,pc2
test: in al,dx
ror al,1
jc test
mov cx,0ffh
here: loop here

mov dx,pb2
mov al,00100000b
add al,bh
out dx,al
mov al,00110000b
add al,bh
out dx,al

mov dx,pa2
in al,dx
ret
konversi endp
;-----

```

```

;-----
bcd proc near          ;procedure untuk ubah BCD

    lea di, buff
    mov cx, 04h
    atas: mov dl, 0ah
    div dl
    mov es:[di+bx], ah
    inc bx
    mov dl, al
    mov ax, 0000h
    mov al, dl
    loop atas
    ret
bcd endp
;-----

;-----
int_64 proc near      ;procedure untuk interrupt-64
    push ax
    push dx
    push si
    push di

    mov al, 00000000b
    mov dx, contregA
    out dx, al
    mov dx, countA0
    in al, dx
    mov ah, al
    in al, dx
    xchg ah, al
    mov dx, 0ffffh
    sub dx, ax
    mov ax, dx
    cmp al, 01h
    jne ubah2
    mov al, 00h

    ubah2: mov bx, 0004h
    call bcd

    mov dx, contregA
    mov al, 00110000b
    out dx, al
    mov al, 0ffh
    mov dx, countA0
    out dx, al
    mov al, 0ffh
    out dx, al

```

```
mov dx,pci
mov al,00h
out dx,al
mov al,00000001b
out dx,al
mov cx,000fh
delayia: loop delayia
mov al,00h
out dx,al
```

```
    mov bh,01h
    mov al,20h
    mov dx,pic0
    out dx,al
```

```
    pop di
    pop si
    pop dx
    pop ax
```

```
iret
```

```
int_64 endp
```

```
;-----
```

```
;-----
```

```
;JMP reset
```

```
@habis: db (@akhir-@habis) dup (0ffh)
```

```
org 020f0h
```

```
@akhir: db 0eah,00h,01h,0f0h,0fbh
```

```
    db 11 dup (0ffh)
```

```
;-----
```



```
{PROGRAM TAMPILAN PADA IBM_PC}  
{DADANG SAMUEL NRP:2912201796}  
{TA-2.PAS}
```

```
uses
```

```
  dos,crt, utility, printer,  
  Graph, display, kompas, hujan, termometer, higo;
```

```
const
```

```
  rx_buffer  = $02f8;  
  div_lsb    = $02f8;  
  div_msb    = $02f9;  
  int_enbl   = $02f9;  
  int_id     = $02fa;  
  line_cont  = $02fb;  
  modem_cont = $02fc;  
  line_stat  = $02fd;  
  modem_stat = $02fe;
```

```
type
```

```
  item = record  
    x,y : byte;  
    pesan : string;  
  end;
```

```
  pilihan = (set_waktu, monitor, cetak_data, keluar);
```

```
var
```

```
  berkastext : text;  
  interval_ku, mulai, sekarang, selang : longint;  
  pilihan_ku : pilihan;  
  array_item : array[0..3] of item;  
  item0, item1, item2, item3 : item;
```

```
  sudahada, terus, tunggu : boolean;  
  ch : char;
```

```
  i,  
  grDriver : Integer;  
  grMode : Integer;  
  ErrCode : Integer;
```

```
  jawab : char;
```

```
  termometer_ku : data_termometer;  
  curah_hujan_ku : curah_hujan;  
  kompas_ku : data_kompas;  
  higo_ku,  
  tekanan_ku,  
  kecepatan_ku : data_higo;  
  kelembaban,  
  tekanan, suhu,  
  kecepatan_angin,  
  curah_hujan_data : real;  
  nilai : string;  
  sudut : real;
```

```
  tgl_mulai, jam_mulai, tgl_selesai, jam_selesai : string;  
  interval : string;  
  namafile : string;
```

```
hari1, bulan1, tahun1, hari2, bulan2, tahun2,  
jam1, menit1, detik1, jam2, menit2, detik2 : word;
```

```
matrik : array[1..24] of byte;
```

```
($I ambildat.pas)
```

```
($I waktu.pas)
```

```
($I cetak.pas)
```

```
procedure init;
```

```
begin
```

```
    item0.x := 39;
```

```
    item0.y := 11;
```

```
    item0.pesan := 'Set Interval';
```

```
    item1.x := 39;
```

```
    item1.y := 12;
```

```
    item1.pesan := 'Monitor';
```

```
    item2.x := 39;
```

```
    item2.y := 13;
```

```
    item2.pesan := 'Cetak Data';
```

```
    item3.x := 39;
```

```
    item3.y := 15;
```

```
    item3.pesan := 'Keluar';
```

```
    array_item[0] := item0;
```

```
    array_item[1] := item1;
```

```
    array_item[2] := item2;
```

```
    array_item[3] := item3;
```

```
    jam_mulai := '';
```

```
    tgl_mulai := '';
```

```
    jam_selesai := '';
```

```
    tgl_selesai := '';
```

```
    interval := '';
```

```
    tunggu := true;
```

```
    namafile := '';
```

```
end;
```

```
function filedata : string;
```

```
begin
```

```
    curon;
```

```
    textbackground(black);
```

```
    clrscr;
```

```
    textbackground(blue);
```

```
    kotak(20,4,59,6);
```

```
    fillkotak(20,4,59,6,' ');
```

```
    write_center(39,4,'Masukkan Nama File Untuk Data',15,blue);
```

```
    filedata := bacakey(22,5,32);
```

```
end;
```

```

procedure hilight(myitem : item);
begin
    write_center(myitem.x, myitem.y, myitem.pesan, 15, cyan);
end;

```

```

procedure dia(myitem : item);
begin
    write_center(myitem.x, myitem.y, myitem.pesan, 15, blue);
end;

```

```

procedure statusline;
begin
    textbackground(red);
    fillkotak(1,22,79,24,' ');
    gotoxy(54,23);
    write('I: ');

    gotoxy(68,23);
    write('F: '+namafile);

    gotoxy(57,23);
    if interval = '' then write('Blm di set')
    else
        write(interval_ku);
end;

```

```

procedure qbr_menu;
begin
    textbackground(black);
    clrscr;
    curoff;
    textcolor(15);
    textbackground(blue);
    kotak(20,4,60,6);
    fillkotak(20,4,60,6,' ');
    gotoxy(30,5);
    write('M E N U      U T A M A');
    kotak(20,10,60,16);
    fillkotak(20,10,60,16,' ');
    write_center(39,11,array_item[0].pesan,15,blue);
    write_center(39,12,array_item[1].pesan,15,blue);
    write_center(39,13,array_item[2].pesan,15,blue);
    write_center(39,15,array_item[3].pesan,15,blue);

    statusline;
end;

```

```

{main}
begin
    curon;
    init;
    qbr_menu;
    terus := true;
    pilihan_ku := set_waktu;
    hilight(array_item[ord(pilihan_ku)]);

```

```

while terus do
begin
  ch := readkey;
  if ch = #0 then
  begin
    ch := readkey;
    case ch of
      #80:
        begin
          if pilihan_ku <> keluar then
          begin
            dim(array_item[ord(pilihan_ku)]);
            pilihan_ku := succ(pilihan_ku);
            hilight(array_item[ord(pilihan_ku)]);
          end;
        end;
      #72:
        begin
          if pilihan_ku <> set_waktu then
          begin
            dim(array_item[ord(pilihan_ku)]);
            pilihan_ku := pred(pilihan_ku);
            hilight(array_item[ord(pilihan_ku)]);
          end;
        end;
    end;
  end;
end;

```

```

if ch = chr(13) then
begin
  bell;
  case pilihan_ku of
    set_waktu :
      begin
        bell;
        waktu;
        gbr_menu;
        hilight(array_item[ord(pilihan_ku)]);
      end;
  end;

```

```

monitor :
begin
  bell;
  namafile := filedata;
  assign(berkastext,namafile);
  ($I-)
  reset(berkastext);
  ($I+);

  sudahada := (ioresult=0);

  if sudahada then
  begin
    jawab := ' ';
    while not((jawab='Y') or (jawab='T')) do

```

```

begin
  write(chr(7));
  gotoxy(20,15);
  write('File Sudah Ada, Hapus file lama ? (y/t)');
  readln(jawab);
  jawab := upcase(jawab);
end;

end;
if jawab='Y' then
begin
  rewrite(berkastext);
  pengawatan(interval_ku);
  closegraph;
  close(berkastext);
end;
curon;
gbr_menu;
hilight(array_item[ord(pilihan_ku)]);
end;
cetak_data :
begin
  bell;
  cetak;
  gbr_menu;
  hilight(array_item[ord(pilihan_ku)]);
end;
keluar :
begin
  terus := false;
end;
end;

end;
end;
curon;
end.

```

```

(DISPLAY.PAS)
unit display;

interface

uses
  Graph,Dos,Crt;

type
  tombol = record
    kiri : integer;
    atas : integer;
    kanan : integer;
    bawah : integer;
    tsize : integer;
    tfont : integer;
    pesan : string;
  end;

  termometer = record
    x,y : integer;
  end;

procedure Button(mytombol : tombol);
procedure Press(mytombol : tombol);
procedure tombol_di_tekan(mytombol : tombol);
procedure light(mytombol : tombol; color : integer);
procedure box( x1, y1, x2, y2, col1, col2 : integer);

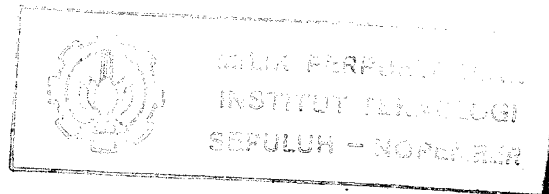
implementation

procedure light(mytombol : tombol; color : integer);
begin
  with mytombol do
  begin
    setcolor(color);
    rectangle(kiri-6, atas-6, kanan+6, bawah+6);
    rectangle(kiri-3, atas-3, kanan+3, bawah+3);
  end;
end;

procedure box(x1,y1,x2,y2,col1,col2 : integer);
begin
  setfillstyle(1,col1);
  bar(x1,y1,x2,y2);
  setfillstyle(1,col2);
  bar(x1+3,y1+3,x2-3,y2-3);
end;

procedure Button(mytombol : tombol);
var
  tlocx, tlocy : integer;

```



```

begin
  with mytombol do
    begin
      setcolor(0);
      rectangle(kiri-1, atas-1, kanan+1, bawah+1);
      setfillstyle(1,7);
      bar(kiri, atas, kanan, bawah);
      setfillstyle(1,15);
      bar(kiri, atas, kanan, atas+3);
      bar(kiri, atas, kiri+4, bawah);
      setfillstyle(1,8);
      bar(kiri, bawah-4, kanan, bawah);
      bar(kanan-4, atas, kanan, bawah);
      setcolor(15);
      line(kiri, bawah, kiri+4, bawah-4);
      line(kiri, atas, kiri, bawah);
      line(kanan-4, atas+3, kanan, atas);
      line(kiri, atas, kanan, atas);
      setfillstyle(1,15);
      floodfill(kiri+1, bawah-3,15);
      floodfill(kanan-3,atas+1,15);
      setttextstyle(tfont,0,tsize);
      setcolor(0);
      tlocx := ((kanan-kiri) div 2 + kiri)-(TextWidth(Pesan) div 2);
      tlocy := ((bawah-atas) div 2 + atas)-(TextHeight(Pesan[1]) div 2);
      outtextxy(tlocx, tlocy, pesan);
      outtextxy(tlocx, tlocy-1, pesan);
      outtextxy(tlocx+1, tlocy, pesan);
      outtextxy(tlocx+1, tlocy-1, pesan);
    end;
  end;

```

```

end;

```

```

procedure Press(mytombol : tombol);

```

```

var

```

```

  tlocx,s, tlocy : integer;

```

```

begin

```

```

  with mytombol do

```

```

    begin

```

```

      setcolor(0);

```

```

      setfillstyle(1,7);

```

```

      bar(kiri, atas, kanan, bawah);

```

```

      setfillstyle(1,8);

```

```

      bar(kiri, atas, kanan, atas+3);

```

```

      bar(kiri, atas, kiri+4, bawah);

```

```

      setfillstyle(1,7);

```

```

      bar(kiri, bawah-4, kanan, bawah);

```

```

      bar(kanan-4, atas, kanan, bawah);

```

```

      setcolor(8);

```

```

      line(kiri, bawah, kiri+4, bawah-4);

```

```

      line(kiri, atas, kiri, bawah);

```

```

      line(kanan-4, atas+3, kanan, atas);

```

```

line(kiri, atas, kanan, atas);
setfillstyle(1,8);
floodfill(kiri+1, bawah-3,8);

floodfill(kanan-3,atas+1,8);
settextstyle(tfont,0,tsize);
setcolor(0);
tlocX := ((kanan-kiri) div 2 + kiri)-(TextWidth(Pesan) div 2);
tlocY := ((bawah-atas) div 2 + atas)-(TextHeight(Pesan[1]) div 2);
outtextxy(tlocX, tlocY, pesan);
outtextxy(tlocX, tlocY-1, pesan);
outtextxy(tlocX+1, tlocY, pesan);
outtextxy(tlocX+1, tlocY-1, pesan);
for s:=100 to 200 do
begin
    sound(s*25);
    delay(1);
    nosound;
end;
end;

end;

procedure tombol_di_tekan(mytombol : tombol);
begin
    with mytombol do
    begin
        press(mytombol);
        button(mytombol);
    end;
end;

end

```



(UTILITY.PAS)

unit utility;

interface

uses

dos, crt;

type

item = record

x,y : byte;

pesan : string;

end;

procedure kotak(x1,y1,x2,y2:byte);

procedure fillkotak(x1,y1,x2,y2:byte; fill : char);

procedure curon;

procedure curoff;

function nilai(s:string) : integer;

procedure write\_center(x,y : byte; s : string; frcol, bkcol : byte);

procedure bell;

function baca(x,y,panjang:byte) : string;

function bacakey(x,y,panjang:byte) : string;

function gettgl(x,y : byte) : string;

function getjam(x,y : byte) : string;

implementation

procedure kotak(x1,y1,x2,y2:byte);

var

i : byte;

begin

for i := x1 to x2 do

begin

gotoxy(i,y1);

write('=');

gotoxy(i,y2);

write('=');

end;

for i := y1 to y2 do

begin

gotoxy(x1,i);

write('|');

gotoxy(x2,i);

write('|');

end;

gotoxy(x1,y1);

write('┌');

gotoxy(x2,y1);

write('┐');

gotoxy(x1,y2);

write('└');

gotoxy(x2,y2);

write('┘');

end;

```
procedure fillkotak(x1,y1,x2,y2:byte; fill : char);
```

```
var
```

```
  i,j : byte;
```

```
begin
```

```
  for i := y1+1 to y2-1 do
```

```
    for j := x1+1 to x2-1 do
```

```
      begin
```

```
        gotoxy(j,i);
```

```
        write(fill);
```

```
      end;
```

```
end;
```

```
procedure curon;
```

```
var
```

```
  reg : Registers;
```

```
begin
```

```
  reg.AH := 1;
```

```
  reg.CH := 0;
```

```
  intr($10,reg);
```

```
  reg.AH := 1;
```

```
  reg.CH := 6;
```

```
  reg.CL := 7;
```

```
  intr($10,reg);
```

```
end;
```

```
procedure curoff;
```

```
var
```

```
  reg : registers;
```

```
begin
```

```
  reg.AH := $1;
```

```
  reg.CH := $20;
```

```
  intr($10,reg);
```

```
end;
```

```
procedure write_center(x,y : byte; s : string; frcol, bkcol : byte);
```

```
var
```

```
  i, len : byte;
```

```
begin
```

```
  len := length(s);
```

```
  gotoxy(x-len div 2,y);
```

```
  textcolor(frcol);
```

```
  textbackground(bkcol);
```

```
  write(s);
```

```
end;
```

```
procedure bell;
```

```
var
```

```
  s : integer;
```

```
begin
```

```
  for s:=100 to 200 do
```

```
    begin
```

```
      sound($25);
```

```
      delay(1);
```

```
      nosound;
```

```
    end;
```

```
end;
```

```

function nilai(s:string) : integer;
var
  i,code : integer;
begin
  val(s,i,code);
  nilai := i;
end;

function baca(x,y,panjang:byte) : string;
var
  i,j : byte;
  l : integer;
  s : string;
  ch : char;
  terus : boolean;

begin
  gotoxy(x,y);
  s := '';
  terus := true;
  i := 0;
  j := 1;
  repeat
    ch := readkey;
    case ch of
      #8 :
        begin
          if i <> 0 then
            begin
              delete(s,i,1);
              i := i-1;
              gotoxy(wherex-1,wherey);
              write(' ');
              gotoxy(wherex-1,wherey);
            end;
          end;
        #27 :
          begin
            terus := false;
            fillkotak(x-1,y-1,x+i,y+1,' ');
            baca := '';
          end;
        #13 :
          begin
            terus := false;
            baca := s;
          end;
        #47..#58 :
          begin
            s := s + ch;
            write(ch);
            i := i+1;
            baca := s;
          end;
    end;
  until (panjang = length(s)) or (terus = false);
end;

```

```

function gettgl(x,y : byte) : string;
var
    tgl, bulan, tahun : string;
begin
    gotoxy(x+2,y);
    write('/ /');
    repeat
        tgl := baca(x,y,2);
    until (nilai(tgl) <= 31 ) and (nilai(tgl) >= 0 );
    repeat
        bulan := baca(x+3,y,2);
    until (nilai(bulan) <= 12) and (nilai(bulan) >=0);
    repeat
        tahun := baca(x+6,y,2);
    until (nilai(tahun) <= 99) and (nilai(tahun) >=0);
    gettgl := bulan + '/' + tgl + '/' + tahun;
end;

```

```

function getjam(x,y : byte) : string;
var
    jam, menit, detik : string;
begin
    gotoxy(x+2,y);
    write(': ');
    repeat
        jam := baca(x,y,2);
    until (nilai(jam) <= 23 ) and (nilai(jam) >= 0 );
    repeat
        menit := baca(x+3,y,2);
    until (nilai(menit) <= 59) and (nilai(menit) >=0);
    repeat
        detik := baca(x+6,y,2);
    until (nilai(detik) <= 59) and (nilai(detik) >=0);

    getjam := jam + ':' + menit + ':' + detik;
end;

```

```

function bacakey(x,y,panjang:byte) : string;
var
    i,j : byte;
    l : integer;
    s : string;
    ch : char;
    terus : boolean;

```

```

begin
    gotoxy(x,y);
    s := '';
    terus := true;
    i := 0;
    j := 1;
    repeat
        ch := readkey;
        case ch of
            #8 :
                begin
                    if i <> 0 then
                        begin

```

```

        delete(s,i,1);
        i := i-1;
        gotoxy(wherex-1,wherey);
        write(' ');
        gotoxy(wherex-1,wherey);
    end;
end;
#27 :
begin
    terus := false;
    fillkotak(x-1,y-1,x+1,y+1,' ');
    bacakey := '';
end;
#13 :
begin
    terus := false;
    bacakey := s;
end;
#32..#255 :
begin
    s := s + ch;
    write(ch);
    i := i+1;
    bacakey := s;
end;
end;
until (panjang=length(s)) or (terus = false);
end;

end

```

{AMBILDAT.PAS}

function gettgl : string;

var

y, m, d, dow : word;

ys, ms, ds : string;

begin

getdate(y,m,d,dow);

str(y,ys);

str(m,ms);

str(d,ds);

gettgl := ds+'/' + ms+'/' + copy(ys,3,2);

end;

function LeadingZero(w : Word) : String;

var

s : String;

begin

Str(w:0,s);

if Length(s) = 1 then

s := '0' + s;

LeadingZero := s;

end;

function ambiljam : string;

var

h, m, s, hund : Word;

begin

GetTime(h,m,s,hund);

ambiljam := LeadingZero(h) + ':' + LeadingZero(m) +

':' + LeadingZero(s);

end;

function in\_data : boolean;

var

h,i,j,k, code : integer;

header, data, dsr, dr, status : byte;

dump : string;

par1, par2, par3, par4, par5, par6 : integer;

begin

port[line\_cont] := \$80;

port[div\_msb] := \$0;

port[div\_lsb] := \$0c;

port[line\_cont] := \$03;

port[int\_id] := \$00;

for i:=0 to 100 do status:=port[rx\_buffer];

repeat

```

port[modem_cont] := $03;
repeat
    i := 0;
    repeat
        status := port[modem_stat];
        dsr    := status AND $20;
        i := i + 1;
    until (dsr = $20) or ( i=1000);

    if i = 1000 then
        begin
(
            write('Periksa hubungan RS 232');
)

            bell;
            in_data := false;
            exit;
        end;

        status := port[line_stat];
        dr     := status AND $01;
    until (dr = $01);

    port[modem_cont] := $00;

    header := port[rx_buffer];

until (header = $05);

for j := 1 to 24 do
begin
    port[modem_cont] := $03;

    repeat
        i := 0;
        repeat
            status := port[modem_stat];
            dsr    := status AND $20;
            i := i + 1;
        until (dsr = $20) or ( i=1000);

        if i = 1000 then
            begin
(
                write('Periksa hubungan RS 232');
)

                bell;
                in_data := false;
                exit;
            end;

            status := port[line_stat];
            dr     := status AND $01;

        until (dr = $01);

```

```

        port[modem_cont] := $00;

        data := port[rx_buffer];

        matrik[j] := data-$30;
    end;

    in_data := true;
end;

procedure tulis_suhu(x,y : integer; suhu : real);
var
    nilai : string;
begin
    str(suhu:3:2, nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+40, nilai+ ' °C');
    outtextxy(x,y+25, 'Suhu');
end;

procedure hapus_suhu(x,y : integer);
begin
    setfillstyle(1,0);
    bar(x-30,y+36,x+30,y+44);
end;

procedure tulis_hujan(x,y : integer; hujan : real);
var
    nilai : string;
begin
    str(hujan:3:2, nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+40, nilai+ ' mm');
    outtextxy(x,y+25, 'Curah Hujan');
end;

procedure tulis_higro(x,y : integer; higro : real);
var
    nilai : string;
begin
    str(higro:3:2, nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+100, nilai+ ' %');
    outtextxy(x,y+85, 'Kelembaban');
end;

```



```

procedure tulis_tekanan(x,y : integer; tekanan : real);
var
    nilai : string;
begin
    str(tekanan:3:2,nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+100,nilai+' mBar');
    outtextxy(x,y+85,'Tekanan');
end;

procedure tulis_kecepatan(x,y : integer; kecepatan: real);
var
    nilai : string;
begin
    str(kecepatan:3:2,nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+100,nilai+' m/s');
    outtextxy(x,y+85,'Kecepatan Angin');
end;

procedure tulis_arah(x,y : integer; arah: real);
var
    nilai : string;
begin
    str(arah:3:2,nilai);
    settextstyle(smallfont,0,5);
    settextjustify(1,1);
    setcolor(11);
    outtextxy(x,y+100,nilai+'°');
    outtextxy(x,y+85,'Arah Angin');
end;

procedure animasi_suhu(mytermometer : data_termometer;
    suhu : real);
var
    panjang : integer;
begin
    with mytermometer do
        begin
            setfillstyle(1,0);
            bar(x-3,y-20,x+3,y-320);
            setfillstyle(1,4);

            panjang := round(suhu);
            if panjang > 300 then panjang := 300;

            bar(x-3,y-20,x+3,y-20-3*panjang);
            hapus_suhu(x,y);

            tulis_suhu(x,y,suhu);
        end;
    end;
end;

```

```

procedure animasi_curah_hujan(mycurah_hujan : curah_hujan;
                               curah_hujan_data:real);
var
    panjang_bar : integer;
begin
    with mycurah_hujan do
        begin
            panjang_bar := round((360/5000)*curah_hujan_data);

            if panjang_bar > 360 then panjang_bar := 360;
            setfillstyle(1,0);
            bar(x-9,y-20,x+9,y-380);
            setfillstyle(1,14);
            bar(x-9,y-20,x+9,y-20-panjang_bar);

            setfillstyle(1,0);
            bar(x-35,y+36,x+35,y+44);
            tulis_hujan(x,y,curah_hujan_data);
        end;
    end;
end;

procedure animasi_kompas(mykompas : data_kompas; arah:real);
var
    sudut : real;
begin
    with mykompas do
        begin
            sudut := (arah-90)*(Pi/180);

            hapus(mykompas);
            panah(mykompas,sudut);

            {
            hapus_suhu(x,y);
            }
            setfillstyle(1,0);
            bar(x-30,y+96,x+30,y+104);

            tulis_arah(x,y,arah);
        end;
    end;
end;

procedure animasi_higro(myhigro : data_higro; kelembaban:real);
var
    sudut : real;
begin
    with myhigro do
        begin
            sudut := (110*Pi/180) + kelembaban*(Pi*320/180)/100;
            hapus_higro(myhigro);
            panah_higro(myhigro,sudut);
            {
            hapus_suhu(x,y);
            }
            setfillstyle(1,0);
            bar(x-30,y+96,x+30,y+104);
            tulis_higro(x,y,kelembaban);
        end;
    end;
end;

```

```

procedure animasi_tekanan(myhigro : data_higro; tekanan:real);
var
    sudut : real;
begin
    with myhigro do
        begin
            sudut := (110*Pi/180) + tekanan*(Pi*320/180)/100;
            hapus_higro(myhigro);
            panah_higro(myhigro,sudut);

            {
                hapus_suhu(x,y);
            }
            setfillstyle(1,0);
            bar(x-35,y+96,x+35,y+104);
            tulis_tekanan(x,y,tekanan+950);
        end;
    end;

procedure animasi_kecepatan(myhigro : data_higro; kecepatan:real);
var
    sudut : real;
begin
    with myhigro do
        begin
            sudut := (110*Pi/180) + kecepatan*(Pi*320/180)/60;
            hapus_higro(myhigro);
            panah_higro(myhigro,sudut);

            {
                hapus_suhu(x,y);
            }
            setfillstyle(1,0);
            bar(x-35,y+96,x+30,y+104);
            tulis_kecepatan(x,y, kecepatan);
        end;
    end;

function ambil_jam_int : longint;
var
    h1, m1, s1, d1 : word;
begin
    gettime(h1,m1,s1,d1);
    ambil_jam_int := (h1*3600 + m1*60 + s1);
end;

procedure pengamatan(selang : longint);
var
    mulai_str, sekarang_str, selisih_str, time_stamp : string;
    selisih : longint;
    tekanan1 : real;
    curah : integer;
begin
    grDriver := Detect;
    InitGraph(grDriver,grMode,'');
    ErrCode := GraphResult;
    if ErrCode <> grOk then
        WriteLn('Graphics error:', GraphErrorMsg(ErrCode))

```

```

else
begin
    termometer_ku.x := 160;
    termometer_ku.y := 338;

    curah_hujan_ku.x := 60;
    curah_hujan_ku.y := 400;

    kompas_ku.x := 285;
    kompas_ku.y := 350;
    kompas_ku.jari_jari := 50;

    higro_ku.x := 500;
    higro_ku.y := 350;
    higro_ku.jari_jari := 50;
    higro_ku.max := 100;
    higro_ku.min := 0;
    higro_ku.interval := 8;

    kecepatan_ku.x := 285;
    kecepatan_ku.y := 150;
    kecepatan_ku.jari_jari := 50;
    kecepatan_ku.max := 60;
    kecepatan_ku.min := 0;
    kecepatan_ku.interval := 10;

    tekanan_ku.x := 500;
    tekanan_ku.y := 150;
    tekanan_ku.jari_jari := 50;
    tekanan_ku.max := 1050;
    tekanan_ku.min := 950;
    tekanan_ku.interval := 8;

    tampilkan_termometer(termometer_ku);
    tampilkan_curah_hujan(curah_hujan_ku);
    tampilkan_kompas(kompas_ku);
    tampilkan_higro(higro_ku);
    tampilkan_higro(tekanan_ku);
    tampilkan_higro(kecepatan_ku);

    settextstyle(sansseriffont,horizdir,4);
    outtextxy(290,5,'WEATHER STATION');

    settextstyle(smallfont,horizdir,4);
    outtextxy(325,45,'by: Dadang S (2912201796)');

    writeln(berkastext,' Waktu':6,' Suhu':8,' Arah':7,
        ' Tekanan':12, ' Kelembaban':13,' Kecepatan':11,
        ' Curah':7);

    writeln(berkastext,gettgl,' angin':14,' hujan':42);
    writeln(berkastext,'(C)':13,'(derajat)':11,'(mBar)':8,
        ' (Z)':11,'(m/s)':12,'(mm)':9);
    writeln(berkastext);

    mulai := ambil_jam_int;

```

```

repeat
  if in_data then
    begin
      suhu := (matrik[4]+matrik[3]*10+matrik[2]*100+matrik[1]*1000);
      suhu := 0.5*suhu;
      sudut := ( matrik[8] + matrik[7]*10 + matrik[6]*100 + matrik[5]*1000 );
      sudut := -0.941575*sudut + 369.061;
      if sudut > 360 then sudut := sudut-360;
      tekanan1 := (matrik[12]+matrik[11]*10+matrik[10]*100+matrik[9]*1000);
      tekanan := (27.30368 + 0.014859*tekanan1)*33.864 -950;
      kelembaban := (matrik[16]+matrik[15]*10+matrik[14]*100+matrik[13]*1000);
      kelembaban := 434.1197-0.39970*kelembaban;
      if kelembaban > 98 then kelembaban := 99;
      kecepatan_angin := (matrik[20]+matrik[19]*10+matrik[18]*100+matrik[17]*1000);
      if kecepatan_angin <> 0 then
        begin
          kecepatan_angin := 1.4644*exp(ln(kecepatan_angin)*0.626492);
        end;
      curah_hujan_data := (matrik[24]+matrik[23]*10+matrik[22]*100+matrik[21]*1000);
      if curah_hujan_data <> 0 then
        begin
          curah_hujan_data := 1.626069 + 0.854609*curah_hujan_data;
        end;
      end
    else
      begin
        bell;
        exit;
      end;

      sekarang := ambil_jam_int;

      selisih := sekarang-mulai;

      tekanan1 := tekanan + 950;

      time_stamp := ambiljam;

      if ((sekarang-mulai) mod selang) = 0 then
        begin
          bell;
          writeln(berkastext, time_stamp:8, suhu:6:2, sudut:9:2,
            tekanan1:10:2, kelembaban:11:2,
            kecepatan_angin:11:2,
            curah_hujan_data:8:2);
        end;

        animasi_suhu(termometer_ku, suhu);
        animasi_curah_hujan(curah_hujan_ku, curah_hujan_data);
        animasi_kompas(kompas_ku, sudut);
        animasi_higro(higro_ku, kelembaban);
        animasi_tekanan(tekanan_ku, tekanan);
        animasi_kecepatan(kecepatan_ku, kecepatan_angin);

      until keypressed;
    end;
  CloseGraph;
end;

```

```

{TERMOMET.PAS}
unit termometer;
interface
uses
Graph,dos,crt;
type
    data_termometer = record
        x,y : integer;
    end;
procedure tampilkan_termometer(mytermometer : data_termometer);
implementation
procedure skala_termometer(x, y, min, max, interval, panjang: integer);
var
    i : integer;
    skala : string;
begin
    i:=0;
    repeat
        str(i*(max-min)div interval + min, skala);
        outtextxy(x-15-textwidth(skala),
            y-(i*panjang) div interval-28, skala);
        i := i + 1;
    until i > interval;

    i:=0;
    repeat
        line(x-11,y-i*panjang div interval-20,x-4,
            y-i*panjang div interval-20);
        line(x+11,y-i*panjang div interval-20,x+4,
            y-i*panjang div interval-20);
        i := i + 1;
    until i > interval;

    i:=0;
    repeat
        line(x-7,y-i*panjang div (interval*4)-20,x-4,
            y-i*panjang div (interval*4)-20);
        line(x+7,y-i*panjang div (interval*4)-20,x+4,
            y-i*panjang div (interval*4)-20);
        i := i + 1;
    until i > interval*4;
end;
procedure tampilkan_termometer(mytermometer : data_termometer);
var
    i : integer;
begin
    with mytermometer do
        begin
            setfillstyle(1,4);
            setcolor(15);
            Circle(x,y-7,8);
            floodfill(x,y-8,15);
            bar(x-4,y-20,x+4,y-1);
            line(x-4,y-320,x-4,y-14);
            line(x+4,y-320,x+4,y-14);
            settextstyle(smallfont,0,5);
            skala_termometer(x,y,0,100,10,300);
        end;
    end;
end
end

```

{WAKTU.PAS}

procedure gbr\_waktu;

begin

```
    curoff;
    textbackground(black);
    clrscr;
    textbackground(blue);
    kotak(15,4,65,16);
    fillkotak(15,4,65,16,' ');
    write_center(39,4,'S E T T I N G   W A K T U',15,blue);
```

```
    gotoxy(18,6);
    write('Jam Mulai      :');
    gotoxy(18,7);
    write('Tanggal Mulai   :');
```

```
    gotoxy(18,9);
    write('Jam Selesai   :');
    gotoxy(18,10);
    write('Tanggal Selesai :');
```

```
    gotoxy(18,12);
    write('Interval      :');
```

```
    gotoxy(50,6);
    write('(jj:mm:dd)');
    gotoxy(50,7);
    write('(hr/bl/th)');
```

```
    gotoxy(50,9);
    write('(jj:mm:dd)');
    gotoxy(50,10);
    write('(hr/bl/th)');
```

```
    gotoxy(50,12);
    write('detik');
```

end;

procedure waktu;

var

ch : char;

begin

```
    curoff;
    textbackground(black);
    clrscr;
    textbackground(blue);
    kotak(20,7,59,9);
    fillkotak(20,7,59,9,' ');
    write_center(39,7,'Masukkan Interval ! (detik)',15,blue);
    curon;
    interval := baca(23,8,8);
    val(interval,interval_ku,i);
    if interval_ku < 10 then interval_ku := 10;
```

end

{DETAK.PAS}

procedure gbr\_cetak;

begin

    curoff;  
    textbackground(black);  
    clrscr;  
    textbackground(blue);  
    kotak(20,4,59,6);  
    fillkotak(20,4,59,6,' ');  
    write\_center(39,4,'Masukkan Nama File Untuk Di Cetak',15,blue);

end;

function siapkan\_printer : boolean;

var

    ch : char;

begin

    curoff;  
    textbackground(black);  
    clrscr;  
    textbackground(blue);  
    kotak(20,7,59,10);  
    fillkotak(20,7,59,10,' ');  
    write\_center(39,8,'Siapkan Printer!',138,blue);  
    write\_center(39,9,'Tekan sembarang tombol (ESC=Exit)',15,blue);  
    ch := readkey;  
    if ch = #27 then siapkan\_printer:= false  
    else  
        siapkan\_printer:= true;

end;

procedure print\_data;

begin

    textbackground(black);  
    clrscr;  
    write('Mencetak Data...');  
    bell;

end;

procedure cetak;

var

    isi, namafile : string;  
    ch : char;  
    ok : boolean;  
    berkasteks : text;



```

begin
  gbr_cetak;
  curon;
  namafile := bacakey(22,5,32);
  if length(namafile) <> 0 then
    begin
      assign(berkasteks,namafile);
      {$I-}
      reset(berkasteks);
      {$I+};
      if ioresult=0 then
        begin
          if siapkan_printer then
            begin
              textbackground(black);
              clrscr;
              write('Mencetak Data...');

              {$I-};
              writeln(lst);
              {$I+}

              if ioresult = 0 then
                begin
                  while not EOF(berkasteks) do
                    begin
                      readln(berkasteks,isi);
                      writeln(lst, isi);
                    end;
                end;
            end;
          close(berkasteks);
        end
      else
        begin
          write('Error');
        end;
    end;
  end;
end

```

```

(KOMPAS.PAS)
unit Kompas;

interface

uses
  Graph,Dos,Crt;

type
  data_kompas = record
    x,y,jari_jari : integer;
  end;

procedure panah(mykompas : data_kompas; sudut:real);
procedure skala_lingkaran(mykompas : data_kompas;
  panjang,interval : integer);

procedure tampilkan_kompas(mykompas : data_kompas);

procedure skala_kompas(mykompas : data_kompas);
procedure hapus(mykompas : data_kompas);

implementation

procedure skala_lingkaran(mykompas : data_kompas;
  panjang,interval : integer);
var
  i, locx1, locy1, locx, locy : integer;
  sudut : real;
begin
  with mykompas do
    begin
      i := 0;
      repeat
        sudut := Pi*i/180;
        locx := round((jari_jari+panjang)*cos(sudut));
        locy := round((jari_jari+panjang)*sin(sudut));

        locx1 := round((jari_jari)*cos(sudut));
        locy1 := round((jari_jari)*sin(sudut));

        line(x+locx1,y+locy1,x+locx,y+locy);
        i := i + interval;
      until i > 360;
    end;
  end;
end;

```

```

procedure skala_kompas(mykompas : data_kompas);
var
  i, locx1, locy1, locx, locy : integer;
  sudut : real;
begin
  with mykompas do
  begin
    skala_lingkaran(mykompas,3,5);

    skala_lingkaran(mykompas,10,45);

    settextstyle(smallfont,0,5);
    setcolor(15);

    locx := round((jari_jari+10)*cos(0))+4;
    locy := round((jari_jari+10)*sin(0))-8;

    outtextxy(x+locx, y+locy,'T');

    locx := round((jari_jari+10)*cos(0.5*Pi))-2;
    locy := round((jari_jari+10)*sin(0.5*Pi));

    outtextxy(x+locx, y+locy,'S');

    locx := round((jari_jari+10)*cos(Pi))-8;
    locy := round((jari_jari+10)*sin(Pi))-8;

    outtextxy(x+locx, y+locy,'B');

    locx := round((jari_jari+10)*cos(1.5*Pi))-2;
    locy := round((jari_jari+10)*sin(1.5*Pi))-16;

    outtextxy(x+locx, y+locy,'U');

    locx := round((jari_jari+10)*cos(0.25*Pi));
    locy := round((jari_jari+10)*sin(0.25*Pi));

    outtextxy(x+locx, y+locy,'TG');

    locx := round((jari_jari+10)*cos(0.75*Pi))-12;
    locy := round((jari_jari+10)*sin(0.75*Pi));

    outtextxy(x+locx, y+locy,'BD');

    locx := round((jari_jari+10)*cos(1.25*Pi))-10;
    locy := round((jari_jari+10)*sin(1.25*Pi))-16;

    outtextxy(x+locx, y+locy,'BL');

    locx := round((jari_jari+10)*cos(1.75*Pi));
    locy := round((jari_jari+10)*sin(1.75*Pi))-16;

    outtextxy(x+locx, y+locy,'TL');
  end;
end;

```

```

procedure hapus(mykompas : data_kompas);
begin
    with mykompas do
    begin
        setcolor(15);
        circle(x,y,jari_jari-1);
        setfillstyle(1,0);
        floodfill(x,y,15);
    end;
end;

procedure panah(mykompas : data_kompas; sudut:real);
var
    locx, locy, locx1, locy1, locx2, locy2:integer;
begin
    with mykompas do
    begin
        setcolor(12);
        circle(x,y,5);
        setfillstyle(1,12);
        floodfill(x,y,12);

        locx := round((jari_jari-2)*cos(sudut));
        locy := round((jari_jari-2)*sin(sudut));

        locx2 := round(15*cos(sudut));
        locy2 := round(15*sin(sudut));

        setcolor(12);

        line(x,y,x+locx,y+locy);

        line(x,y,x-locx2,y-locy2);

        locx1 := round((jari_jari-25)*cos(sudut-0.088));
        locy1 := round((jari_jari-25)*sin(sudut-0.088));

        line(x+locx,y+locy,x+locx1,y+locy1);

        locx1 := round((jari_jari-25)*cos(sudut+0.088));
        locy1 := round((jari_jari-25)*sin(sudut+0.088));

        line(x+locx,y+locy,x+locx1,y+locy1);
    end;
end;

procedure tampilkan_kompas(mykompas : data_kompas);
begin
    with mykompas do
    begin
        setcolor(15);
        circle(x,y,jari_jari-1);
        circle(x,y,jari_jari);
        skala_kompas(mykompas);
    end;
end;
end

```

(HIGRO.PAS)

unit higro;

interface

uses

Graph, crt;

type

data\_higro = record  
    x : integer;  
    y : integer;  
    jari\_jari : integer;  
    max : word;  
    min : word;  
    interval : word;  
end;

procedure skala\_lingkaran\_higro(myhigro : data\_higro; panjang, selang : integer);  
procedure skala\_higro(myhigro : data\_higro);  
procedure hapus\_higro(myhigro : data\_higro);  
procedure panah\_higro(myhigro : data\_higro; sudut:real);  
procedure tampilkan\_higro(myhigro : data\_higro);

implementation

procedure skala\_lingkaran\_higro(myhigro : data\_higro;  
    panjang, selang : integer);

var

i, locx1, locy1, locx, locy : integer;  
interval\_sudut : real;

begin

    with myhigro do

    begin

        i := 0;

        repeat

            interval\_sudut := (110\*Pi/180) + i\*(Pi\*320/180)/selang;

            locx := round((jari\_jari+panjang)\*cos(interval\_sudut));

            locy := round((jari\_jari+panjang)\*sin(interval\_sudut));

            locx1 := round((jari\_jari)\*cos(interval\_sudut));

            locy1 := round((jari\_jari)\*sin(interval\_sudut));

            line(x+locx1,y+locy1,x+locx,y+locy);

            i := i + 1;

        until i > selang;

    end;

end;

```

procedure skala_higro(myhigro : data_higro);
var
  i, locx1, locy1, locx, locy : integer;
  interval_sudut : real;
  skala : string;
  radius : integer;
  jarak : real;

begin
  with myhigro do
    begin
      radius := jari_jari + 25;

      setttextjustify(1,1);

      skala_lingkaran_higro(myhigro,3,72);

      skala_lingkaran_higro(myhigro,10,interval);

      setttextstyle(smallfont,0,5);
      setcolor(15);

      i := 0;
      repeat
        jarak := i*(max-min) / interval + min;
        str(jarak:0:1 , skala);
        interval_sudut := (110*Pi/180) + i*(Pi*320/180)/interval;
        locx := round( radius *cos(interval_sudut));
        locy := round( radius *sin(interval_sudut));

        outtextxy(x+locx, y+locy, skala);
        i := i + 1;

      until i > interval;

      setttextjustify(0,2);

    end;

  end;

procedure hapus_higro(myhigro : data_higro);
begin
  with myhigro do
    begin
      setcolor(15);
      circle(x,y,jari_jari-1);
      setfillstyle(1,0);
      floodfill(x,y,15);

    end;

  end;

```

```

procedure panah_higro(myhigro : data_higro; sudut:real);
var
  locx, locy, locx1, locy1, locx2, locy2:integer;
begin
  with myhigro do
    begin
      setcolor(12);
      circle(x,y,5);
      setfillstyle(1,12);
      floodfill(x,y,12);

      locx := round((jari_jari-2)*cos(sudut));
      locy := round((jari_jari-2)*sin(sudut));

      locx2 := round(15*cos(sudut));
      locy2 := round(15*sin(sudut));

      setcolor(12);

      line(x,y,x+locx,y+locy);

      line(x,y,x-locx2,y-locy2);

      locx1 := round((jari_jari-25)*cos(sudut-0.088));
      locy1 := round((jari_jari-25)*sin(sudut-0.088));

      line(x+locx,y+locy,x+locx1,y+locy1);

      locx1 := round((jari_jari-25)*cos(sudut+0.088));
      locy1 := round((jari_jari-25)*sin(sudut+0.088));

      line(x+locx,y+locy,x+locx1,y+locy1);
    end;
  end;
end;

```

```

procedure tampilkan_higro(myhigro : data_higro);
begin
  with myhigro do
    begin
      setcolor(15);
      circle(x,y,jari_jari-1);
      circle(x,y,jari_jari);
      skala_higro(myhigro);
    end;
  end;
end.

```

{HUJAN.PAS}

```
unit hujan;
interface
uses
  Graph,Dos,Crt;
type
  curah_hujan = record
    x,y : integer;
  end;
procedure tampilkan_curah_hujan(mycurah_hujan: curah_hujan);
implementation

procedure skala_curah_hujan(x,y:integer; min, max, interval, panjang: word);
var
  i : integer;
  skala : string;
begin
  i:=0;
  repeat
    str((i*(max-min) div interval + min), skala);
    outtextxy(x-19-textwidth(skala),
      y-(i*panjang) div interval-23, skala);
    i := i + 1;
  until i > interval;

  i:=0;
  repeat
    line(x-17,y-i*panjang div interval-20,x-10,
      y-i*panjang div interval-20);
    line(x+17,y-i*panjang div interval-20,x+10,
      y-i*panjang div interval-20);
    i := i + 1;
  until i > interval;

  i:=0;
  repeat
    line(x-12,y-i*panjang div (interval*8)-20,x-10,
      y-i*panjang div (interval*8)-20);
    line(x+12,y-i*panjang div (interval*8)-20,x+10,
      y-i*panjang div (interval*8)-20);
    i := i + 1;
  until i > interval*8;

  i:=0;
  repeat
    line(x-15,y-i*panjang div (interval*4)-20,x-10,
      y-i*panjang div (interval*4)-20);
    line(x+15,y-i*panjang div (interval*4)-20,x+10,
      y-i*panjang div (interval*4)-20);
    i := i + 1;
  until i > interval*4;

  line(x-10,y-16,x-10,y-20-panjang);
  line(x+10,y-16,x+10,y-20-panjang);

end;
```



```

procedure tampilkan_curah_hujan(mycurah_hujan: curah_hujan);
var
  i, tsize, tfont : integer;
begin
  with mycurah_hujan do
    begin
      line(x-30,y,x+30,y);
      line(x-30,y,x-30,y-8);
      line(x-30,y-8,x-10,y-16);
      line(x-10,y-16,x-10,y-20);
      line(x-10,y-20,x+10,y-20);
      line(x+10,y-20,x+10,y-16);
      line(x+10,y-16,x+30,y-8);
      line(x+30,y-8,x+30,y);

      (      line(x-10,y-16,x-10,y-380);
        line(x+10,y-16,x+10,y-380);
      )

      settextstyle(smallfont,0,5);
      skala_curah_hujan(x,y,0,5000,8,360);
    end;
  end;

end

```

25 NOV 1993

JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI-ITS

EE 1799 TUGAS AKHIR

Nama Mahasiswa : Dadang Samuel  
Nrp : 2912201796  
Bidang Studi : Elektronika  
Dosen Wali : Ir. Soetikno  
Dosen pembimbing : 1. Ir. Soetikno  
2. Ir. Djoko Purwanto  
Judul Tugas Akhir :

INTERFACE AKUISISI 6 PARAMETER DATA METEOROLOGI DENGAN MINIMUM  
SISTEM 8088 YANG DIHUBUNGKAN PADA KOMPUTER IBM PC SECARA SERI

Uraian Tugas Akhir:

Dari kenyataan bahwa data meteorologi merupakan data yang sangat penting bagi dunia pertanian, penerbangan, pelayaran maka diperlukan alat untuk melakukan pendataan yang optimal, dimana data yang ada akan diproses lebih lanjut guna menyimpulkan akan arti dari parameter meteorologi tersebut.

Prinsip utama kerja alat ini adalah merubah 6 parameter seperti kelembapan, tekanan udara, curah hujan, kecepatan angin, arah angin dan suhu udara menjadi besaran listrik oleh masing-masing transdusernya yang akan diubah menjadi besaran digital dan secara multipleks data diakuisisi oleh pengontrol utama sistem minimum 8088 dimana data akan ditransfer secara seri melalui RS232 ke dalam IBM PC yang digunakan untuk media tampilan dan penyimpanan data pada disket.

Surabaya, 1 Nopember 1993

Dosen Pembimbing I

(Ir. Soetikno)  
NIP. 130445231

Dosen Pembimbing II

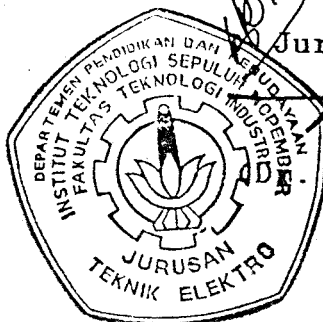
(Ir. Djoko Purwanto)  
NIP. 131879397

Menyetujui  
Bidang Studi Elektronika  
Koordinator

22/93 (Ir. Soetikno)  
NIP. 130445231

Mengetahui  
Jurusan Teknik Elektro FTI-ITS  
Ketua

29/93  
Ir. Moch Salehudin, M.Eng.Sc)  
NIP. 130532026



## USULAN TUGAS AKHIR

- B. JUDUL TUGAS AKHIR : INTERFACE AKUISISI 6 PARAMETER DATA METEOROLOGI DENGAN MINIMUM SISTEM 8088 YANG DIHUBUNGKAN PADA KOMPUTER IBM PC SECARA SERI.
- C. RUANG LINGKUP : -MIKROELEKTRONIKA  
-INSTRUMENTASI ELEKTRONIKA  
-RANGKAIAN LINIER AKTIF  
-RANGKAIAN TAK LINIER TAK AKTIF  
-PENGUKURAN ELEKTRONIKA  
-BAHASA ASSEMBLY
- D. LATAR BELAKANG : Dari arti penting data meteorologi, adanya pendataan yang konvensional dirasa sangat tidak efektif dan efisien. Dengan alat ini diharapkan permasalahan diatas dapat diatasi sehingga didapat hasil yang optimal, dimana data tersebut akan diolah untuk diartikan dan disimpulkan.
- D. PERMASALAHAN : Data akuisisi ini direncanakan dapat mendata dengan selang waktu tertentu dengan ketelitian dan kecepatan yang optimal dibanding sistim konvensional yang ada.
- E. BATASAN MASALAH : Data akuisisi ini direncanakan dengan sistem minimum 8088 dengan IBM PC

sebagai alat penampil, penyimpanan data untuk diproses lebih lanjut.

- F. LANGKAH-LANGKAH : -Mempelajari parameter-parameter meteorologi yang penting.
- Mempelajari minimum sistem 8088 dan teknik interfacing dengan IBM PC.
- Mempelajari transduser yang diperlukan.
- Perencanaan dan pembuatan alat beserta perangkat lunaknya.
- Pembahasan dan penyelesaian tugas akhir.

G. RELEVANSI : Akuisisi data ini diharapkan dapat memberikan sumbangan pemikiran pada jawatan meteorologi sebagai alternatif untuk dikembangkan lebih lanjut.

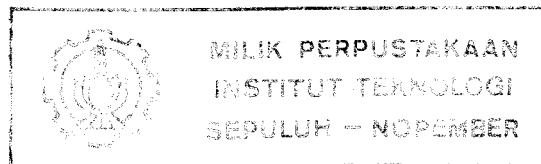
H. JADWAL KEGIATAN :

BULAN KE KEGIATAN	I	II	III	IV	V	VI
1. Studi literatur	xxx	xxx	xxx	xxx		
2. Analisa masalah		xxx	xxx	xxx		
3. Perangkat keras		xxx	xxx	xxx	xxx	
4. Perangkat lunak			xxx	xxx	xxx	
5. Percobaan					xxx	xxx
6. Penyelesaian				xxx	xxx	xxx

## DAFTAR RIWAYAT HIDUP



DADANG SAMUEL dilahirkan pada tanggal 18 November 1968 di Pekalongan, anak terakhir dari dua bersaudara. Ayah bernama MS. Samiarjo dan ibu bernama Murdiati.



### RIWAYAT PENDIDIKAN

1. TK Kutilang, lulus tahun 1974
2. SDN I Pekalongan, lulus tahun 1980
3. SMPN I Pekalongan, lulus tahun 1983
4. SMAN I Pekalongan, lulus tahun 1986
5. Politeknik Bandung, lulus tahun 1989
6. Masuk Teknik Elektro, FTI ,ITS tahun 1991 melalui program lintas jalur, dan diharapkan lulus pada ujian Tugas Akhir periode Wisuda September 1994.

### PENGALAMAN KERJA

1. Sebagai karyawan di PT. SCHLUMBERGER GEOPHYSICS NUSANTARA selama 2 tahun (November 1989 - September 1991).

Selama belajar di ITS, penulis pernah aktif menjadi asisten praktikum Rangkaian Listrik dan Elektronika.